

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Les couches supérieures de l'ISO

#### Compatibilité avec différents types de réseau. Projet d'implémentation

Henrion, Philippe; Waltzing, Michel

*Award date:*  
1983

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES  
UNIVERSITAIRES  
N.D. DE LA PAIX

**NAMUR**



---

INSTITUT D'INFORMATIQUE

LES COUCHES SUPERIEURES DE L'ISO

COMPATIBILITÉ AVEC DIFFÉRENTS  
TYPES DE RÉSEAU

PROJET D'IMPLÉMENTATION

PROMOTEUR: J. BRUNIN

MÉMOIRE PRÉSENTÉ PAR  
HENRION PHILIPPE  
WALTZING MICHEL  
EN VUE DE L'OBTENTION DU GRADE DE  
LICENCE ET MAÎTRE EN INFORMATIQUE

ANNÉE ACADEMIQUE 1982-1983



Il nous tient à coeur de remercier les personnes  
qui, par leur aide, leur patience et leurs conseils,  
nous ont permis de mener à bien ce travail.

Nous remercions plus particulièrement:

Monsieur BRUNIN: promoteur du mémoire,

Mademoiselle SCOYER: lors de la réalisation pratique  
de la maquette.

## A V A N T   P R O P O S

=====

Les applications de l'informatique étaient jusqu'à présent perfectionnées isolément. La tendance actuelle ( $\pm$  10 ans) est à leur interconnexion. La possibilité de faire "fonctionner" des réseaux connectant plusieurs dizaines de terminaux pouvant "dialoguer" entre eux, confère à la technique de transmission de données, une place dans les sujets d'étude future.

Les développements de la technologie des réseaux illustrent une des tendances fondamentales de l'informatique aujourd'hui: le rôle de plus en plus important du logiciel. Il est certain que les réseaux quels que soient leur configuration, seront appelés à relier des équipements parfois difficilement compatibles. Les problèmes de la communication entre systèmes informatiques prennent une autre dimension, dès que l'on se propose de mettre en relation des machines "non compatibles". Cela impose, dès lors, la mise en place de réseaux adaptés aux ordinateurs, c'est-à-dire qui permettent à partir d'un nombre limité de raccordements d'atteindre tous les autres types de machines.

Avec le mûrissement des réseaux, l'idée d'une structuration en couches des systèmes de communication s'est avérée essentielle. Les organismes internationaux, tels que l'ISO par exemple ont formulé différents projets la concernant.

Ces diverses propositions étant relativement abstraites et générales, les principaux constructeurs d'ordinateurs étudiant leur propre réseau ont, dès lors la possibilité de les interpréter à leur convenance. Ceci les conduit à des



discordances, parfois voulues, qui rendent les réseaux développés difficilement compatibles à l'heure actuelle.

Parallèlement aux travaux de ces constructeurs, des organismes internationaux, tels que le CCITT et ECMA, essayent de concrétiser ces propositions à leurs niveaux.

L'architecture adoptée par l'ISO sous le nom de RM/OSI (reference model of open systems interconnection) est basée sur ce concept de couches. Dans cette approche, une structure complexe est portionnée en un nombre de niveaux "indépendants" chacun remplissant certaines fonctions. Ces niveaux sont au nombre de sept. Les trois couches inférieures (1, 2, 3) sont dépendantes des réseaux considérés. Cependant, comme chacun de ceux-ci a ses caractéristiques propres, il est souhaitable de conserver intacte l'indépendance des couches supérieures de l'architecture vis-à-vis des couches inférieures.

Ces différents points sont examinés dans la suite de ce projet. Celui-ci est composé de deux parties distinctes:

- La première consiste en une étude "théorique" des couches supérieures, à savoir les couches Transport, Session (transmission de texte), Présentation et Application. Comme l'objectif est de rendre ces quatre couches supérieures les plus indépendantes possible du type de réseau utilisé (X.25, X.21, Ethernet...) il est souhaitable d'introduire une couche supplémentaire qui va servir d'interface. Celle-ci est étudiée pour un alignement des différents réseaux sur X.25. On aborde également un moyen de faire transiter une information standardisée d'une couche à l'autre.
- La seconde est une maquette visant à simuler une solution minimale dans le cas d'un réseau local (Pas de circuit virtuel, nombre limité de stations). Dans

cette solution ne sont prises en considération que la couche utilisateur (niveau 7), l'interface "banalisé" et le moniteur gérant la suite des opérations. L'introduction des couches intermédiaires (transport, session, présentation) y est également abordée.



# TABLE DES MATIERES

=====

PREMIERE PARTIE . . . . .	1
<u>CHAPITRE I</u> - NOTIONS DE BASE . . . . .	3
1.1 - PROTOCOLES ET PROCÉDURE . . . . .	3
1.2 - PROCESSUS . . . . .	6
1.2.1 - UNITE CONTROLE . . . . .	6
1.2.2 - PROCESSEUR . . . . .	6
1.2.3 - PROCESSUS . . . . .	6
1.3 - JUSTIFICATION DU DÉCOUPAGE DE L'ARCHITECTURE EN COUCHES . . . . .	7
1.3.1 - COUCHES DES RESEAUX . . . . .	8
- Couche 3: Acheminement . . . . .	8
- Couche 2: Liaison . . . . .	8
- Couche 1: Physique . . . . .	9
1.3.2 - COUCHES D'APPLICATION . . . . .	9
- Couche 7: Application . . . . .	9
- Couche 6: Présentation . . . . .	10
- Couche 5: Session . . . . .	11
- Couche 4: Transport . . . . .	12
1.4 - PROCESSUS À COUCHES HIÉRARCHISÉES . . . . .	14
1.4.1 - PRINCIPE DE LA STRUCTURATION EN COUCHES . . . . .	14
1.4.2 - FONCTIONS GENERALES DES COUCHES . . . . .	17
1.4.2.1 - Niveau 4 Transport . . . . .	17
1.4.2.2 - Niveau 5 Session . . . . .	21
1.4.2.3 - Niveau 6 Présentation . . . . .	24

1.5 - LE NIVEAU TRANSPORT . . . . .	28
<u>CHAPITRE II</u> - LES PROTOCOLES DE LA COUCHE 4 ET D'INTERCOUCHES . . . . .	40
2.1 - PROTOCOLE BSC . . . . .	40
2.2 - LES PROTOCOLES DU NIVEAU 4 ET D'INTERCOUCHES .	43
<u>CHAPITRE III</u> - LES PRIMITIVES D'INTERCOUCHES . . . .	47
<u>CHAPITRE IV</u> - FONCTIONNEMENT GÉNÉRAL . . . . .	53
4.1 - PROGRESSION INTERCOUCHES DES PRIMITIVES . . . .	53
4.1.1 - PROGRESSION DE TYPE A . . . . .	54
4.1.2 - PROGRESSION DE TYPE B . . . . .	57
1. Appel . . . . .	57
2. Libération . . . . .	62
3. Données . . . . .	62
4. Pseudo-données . . . . .	67
4.2 - LES OCTETS D'INFORMATION ET LEUR CHEMINEMENT .	70
4.2.1 - INTRODUCTION . . . . .	70
4.2.2 - OCTETS D'INFORMATION EN ENTREE (BIU/IR) ET EN SORTIE (BOU/7) . . . . .	72
4.2.3 - OCTETS D'INFORMATION A L'ARRIVEE (BINR/7) ET AU DEPART (BONR/IR) . . . . .	77
4.2.4 - PROGRAMMES DE COUCHES . . . . .	85
4.2.5 - MECANISMES DES OPERATIONS . . . . .	88
4.3 - IMBRICATION DES OPÉRATIONS . . . . .	94
4.3.1 - SEQUENCE DE BASE DES OPERATIONS . . . . .	94
4.3.2 - ILLUSTRATIONS DES SEQUENCES . . . . .	97



<u>CHAPITRE V - FONCTIONNEMENT PARTICULIER DES COUCHES.</u>	105
INTRODUCTION . . . . .	105
A - GENERALITES . . . . .	105
B - FONCTIONS DES DIFFERENTES COUCHES SUPERIEURES . . . . .	107
5.1 - NIVEAU TRANSPORT <sup>(4)</sup> . . . . .	109
5.1.1 - FONCTIONS GENERALES . . . . .	109
5.1.2 - FONCTIONS DETAILLEES . . . . .	114
5.1.3 - CONTROLE DE FLUX . . . . .	119
5.1.3.1 - Principes . . . . .	119
5.1.3.2 - Contrôle de flux . . . . .	122
5.1.4 - CONTROLE DE BOUT EN BOUT . . . . .	126
5.1.5 - FONCTIONS SPECIFIQUES DU PROTOCOLE . .	127
5.2 - NIVEAU SESSION <sup>(5)</sup> . . . . .	137
5.2.1 - FONCTION GENERALE . . . . .	137
5.2.2 - FONCTIONS DETAILLEES DU PROTOCOLE . .	137
5.3 - NIVEAU PRÉSENTATION <sup>(6)</sup> . . . . .	145
5.3.1 - IMPORTANCE DE CETTE COUCHE . . . . .	145
5.3.2 - FONCTIONS GENERALES . . . . .	151
5.3.3 - FONCTIONS DETAILLEES DES PROTOCOLES POSSIBLES . . . . .	153
5.3.3.1 - Classe 0 . . . . .	153
5.3.3.2 - Classe I . . . . .	155
5.3.3.3 - Classe II . . . . .	158
5.3.3.4 - Classe III . . . . .	168
5.3.3.5 - Classe IV . . . . .	173
<sup>NIVEAU</sup> 5.4 - APPLICATION <sup>(7)</sup> . . . . .	178
5.4.1 - PRIMITIVES DU NIVEAU 7 . . . . .	179
5.4.2 - FORMATS . . . . .	180

<u>CHAPITRE VI - INTERFACE BANALISE . . . . .</u>	184
6.1 - COMPATIBILITÉ ENTRE RÉSEAUX . . . . .	184
6.1.1 - INTERFACAGE ENTRE RESEAUX . . . . .	184
6.1.1.a - au niveau du terminal . . . . .	184
6.1.1.b - au niveau du réseau . . . . .	187
6.2 - PROTOCOLE DE BASE ALIGNÉ SUR X.25 ET LES PRIMITIVES D'INTERCOUCHES . . . . .	194
6.2.1 - INTRODUCTION . . . . .	194
6.2.2 - SELECTION DES PRIMITIVES X.25 . . . . .	195
6.3 - CONVERSION DES PRIMITIVES X.25 ET OCTETS D'ACTIONS (ET INVERSEMENT) . . . . .	201
6.3.1 - PRINCIPE . . . . .	201
6.3.2 - L'OCTET D'INFORMATION (RAPPEL) . . . . .	206
6.3.3 - L'INTERFACE BANALISE . . . . .	208
6.3.3.1 - Objectifs . . . . .	208
6.3.3.2 - Principe d'interfaçage . . . . .	209
6.4 - ALIGNEMENT DE X.21 SUR X.25 . . . . .	212
6.5 - ALIGNEMENT D'ETHERNET SUR X.25 . . . . .	216
6.6 - ALIGNEMENT ASYNCHRONE SUR X.25 . . . . .	220
6.7 - ALIGNEMENT DES PROTOCOLES . . . . .	222
<u>CHAPITRE VII - COMPLÉMENTS . . . . .</u>	227
7.1 - INTRODUCTION . . . . .	227
7.2 - FONCTIONNEMENT MULTI-PROCESSEURS . . . . .	229
7.2.1 - ETAT D'UN PROCESSUS . . . . .	229
7.2.2 - FONCTIONNEMENT GENERAL ET PHYSIQUE EN MODE MONO-PROCESSEUR . . . . .	230



7.2.2.1 - Principes . . . . .	230
7.2.2.2 - Opérations de sorties . . .	232
7.2.2.3 - Opérations d'entrée . . . .	236
7.2.3 - FONCTIONNEMENT GENERAL EN MODE MULTI- PROCESSEURS . . . . .	238
7.2.3.1 - Les interruptions . . . . .	238
- Interruptions entrantes	
- Hardware . . . . .	238
- Software . . . . .	238
- Interruptions sortantes	
- Hardware . . . . .	240
- Software . . . . .	240
7.2.3.2 - Plusieurs processeurs à niveaux . . . . .	241
- Interruptions . . . . .	241
- Instructions spécialisées	242
- Synchronisation . . . . .	243
7.2.3.3 - Cas de plusieurs processus.	244
7.3 - SESSION ET ÉCHANGES INTERACTIFS . . . . .	246
7.3.1 - APPLICATION . . . . .	246
7.3.2 - ENSEMBLE DES PRIMITIVES DE SESSION INTERACTIVE . . . . .	246
7.3.3 - DIAGRAMMES D'ETATS . . . . .	248
7.3.4 - IMPLEMENTATION DES FONCTIONS PROPOSEES POUR LA SESSION . . . . .	248
7.3.5 - INTERACTIF SUR RCP . . . . .	251
7.4 - REPRISE ET RÉINITIALISATION . . . . .	253
7.4.1 - RESET OU REINITIALISATION . . . . .	253
7.4.2 - RESTART OU REPRISE . . . . .	254
7.5 - RÉDUCTION DE L'ARCHITECTURE . . . . .	256

DEUXIEME PARTIE . . . . .	258
1 - OBJECTIF DE TRAVAIL . . . . .	260
2 - HYPOTHÈSES DE TRAVAIL . . . . .	262
3 - SPÉCIFICATIONS . . . . .	264
1 - INTRODUCTION . . . . .	264
2 - APPEL . . . . .	266
3 - DONNEES . . . . .	268
4 - LIBERATION . . . . .	270
5 - TABLES SEQUENTIELLES . . . . .	273
4 - CHOIX LIÉS À LA MAQUETTE . . . . .	274
5 - INTRODUCTION DES COUCHES 4, 5 ET 6 . . . . .	276
6 - DISCUSSION DES MODIFICATIONS DE LA MAQUETTE . . .	286
CONCLUSIONS . . . . .	287
BIBLIOGRAPHIE . . . . .	289

## TABLE DES MATIÈRES DES ANNEXES

### ANNEXE A:

DESCRIPTION GÉNÉRALE ET RÈGLES DE FONCTIONNEMENT APPLICABLES AUX DIAGRAMMES DES ÉTATS, . . . . .	1
---	---

### ANNEXE B:

GRAPHES D'ÉTATS DU NIVEAU TRANSPORT, . . . . .	4
--	---

### ANNEXE C:

GRAPHES D'ÉTATS DU NIVEAU SESSION, . . . . .	17
--	----

### ANNEXE D:

FONCTION DE PRÉSENTATION DU CODE ASCII ET ASCII ÉTENDU, . . . . .	27
---	----

### ANNEXE E:

PRIMITIVES ET DIAGRAMMES D'ÉTATS DU NIVEAU PRÉSENTATION, . . . . .	47
--	----

### ANNEXE F:

PROTOCOLE D'APPAREIL VIRTUEL, . . . . .	57
---	----

### ANNEXE G:

PROTOCOLE STANDARD X.25, . . . . .	71
------------------------------------	----

### ANNEXE H:

LA PROCÉDURE X.21 ET SON ALIGNEMENT SUR X.25, . . . . .	81
---	----

### ANNEXE I:

ASSEMBLEUR ~ DÉSASSEMBLEUR DE PAQUETS: ADP, . . . . .	90
---	----

### ANNEXE J:

AMANDÉ RECOMMENDATIONS X.28 AND X.29, . . . . .	110
---	-----



P R E M I È R E   P A R T I E

=====



## CHAPITRE I - NOTIONS DE BASE

=====

### 1.1 - PROTOCOLES ET PROCÉDURES

### 1.2 - PROCESSUS

1.2.1 - Unité contrôle

1.2.2 - Processeur

1.2.3 - Processus

### 1.3 - JUSTIFICATION DU DÉCOUPAGE DE L'ARCHITECTURE EN COUCHES

1.3.1 - Couches des réseaux

- Couche 3: Acheminement
- Couche 2: Liaison
- Couche 1: Physique

1.3.2 - Couches d'application

- Couche 7: Application
- Couche 6: Présentation
- Couche 5: Session
- Couche 4: Transport

### 1.4 - PROCESSUS À COUCHES HIÉRARCHISÉES

1.4.1 - Principe de la structuration en couches

1.4.2 - Fonctions générales des couches

1.4.2.1 - Niveau 4 Transport

1.4.2.2 - Niveau 5 Session

1.4.2.3 - Niveau 6 Présentation

### 1.5 - LE NIVEAU TRANSPORT

# CHAPITRE I

---

## NOTIONS DE BASE

### 1.1 - PROTOCOLES ET PROCÉDURES

Pour établir une communication, que ce soit entre deux êtres humains ou deux machines, il faut suivre un certain nombre de règles, de lois. L'exemple du téléphone est intéressant à plus d'un titre.

Nous aurons pour une conversation téléphonique, par exemple:

- établissement de la liaison
  - décrocher le combiné
  - attendre la tonalité
  - former le numéro
  - ...
- dialogue
  - reconnaissance des interlocuteurs
  - ...
- etc

Dans le cadre de la communication entre "humains", les règles et les lois sont informelles et complexes. Pour éviter ces inconvénients, il faut dans le cadre de communication entre ordinateurs, que ces règles soient précises, simples en un mot: formelles.



Pour effectuer la synchronisation de deux systèmes tournant ensemble, on a recourt à l'échange de messages de coordination. Celui-ci suit scrupuleusement des procédures établies appelées "protocole".

Un protocole est une description des procédures d'échange d'informations entre processus faisant partie d'un réseau d'interconnexion (local ou distribué) et ayant des fonctions très précises à remplir. Il s'agit, par exemple, d'assurer le recouvrement des erreurs lors du transfert d'informations pour assurer la validité de celles-ci ou encore de gérer et de contrôler les échanges d'informations entre processus.

Les fonctions d'un protocole sont assurées par l'échange de messages entre les processus. Le format et la signification de ces messages constituent la définition logique ou syntaxique du protocole.

La procédure détermine quant à elle les règles d'utilisation séquentielles de communication de messages pour une transmission donnée. La procédure sera déterminée pour une séquence d'actions bien définies telle que : appel, liaison de données, libération, reprise, contrôle...

Parmi les caractéristiques principales des protocoles, on peut encore citer : le contrôle d'erreurs sur les messages échangés, le contrôle de flux des messages, le contrôle des congestions et les stratégies de routage.

Le contrôle d'erreurs permet de conserver l'intégrité des données de l'utilisateur ainsi que les messages de contrôle.

Le contrôle de flux permet de réguler le trafic du générateur de messages, en fonction de la capacité de traitement et/ou de mémorisation du destinataire.

Le contrôle de flux, ainsi d'ailleurs que le contrôle de congestion, permet de ne pas saturer le sous-réseau de transport et d'assurer ainsi un service satisfaisant à un grand nombre d'utilisateurs.

Le contrôle de séquençement porte sur une suite numérotée normale.

Le contrôle du routage permet également d'assurer un meilleur service au niveau du sous-réseau, en permettant d'éviter les noeuds surchargés (situation de congestion) ou défectueux.



## 1.2 - PROCESSUS

- 1 - Unité centrale ou C P U (Central Processing Unit): partie d'un ordinateur qui assure l'interprétation et l'exécution des instructions.
- 2 - Processeur: organe effectuant un traitement plus ou moins élaboré. Il peut se composer pour ce faire d'une unité centrale (CPU), de registres et même de mémoires. Exemple: processeur d'entrée/sortie évoquant la notion de canal (CPU, registres, rapides, mémoire ROM).
- 3 - Processus. Les protocoles sont implantés par l'intermédiaire de processus entre lesquels s'échangent des messages suivant les règles définies par les premiers.

Le message est défini par son format, sa signification et sa procédure. Un processus est un processeur réel ou virtuel, sollicité par un message venant d'un processus extérieur à travers une porte d'accès. Un processus étant de nature séquentielle, son comportement est fonction de son état et celui-ci sera modifié à tout moment en fonction du message incident. Le résultat du calcul effectué sur le message d'entrée pour le nouvel état est envoyé vers un processus extérieur au travers d'une porte de sortie et la nature de ce message déterminera pour ce dernier un nouvel état.

Ce dialogue se prolongera jusqu'à la fin du traitement ou une éventuelle rupture de séquence.

### 1.3 - JUSTIFICATION DU DÉCOUPAGE DE L'ARCHITECTURE EN COUCHES

De par sa nature même, en téléinformatique, il existe d'une part, une ou plusieurs "applications" d'utilisateurs différents et de nature diverses, destinées à communiquer entre elles à distance. D'autre part, au moins un "réseau de transmission de données" permet cette communication.

La raison première de l'introduction de couches dans les logiciels de téléinformatique provient du fait qu'il était intéressant pour l'exploitation, de tamponner l'espace "Applications-Réseaux", de telle sorte que la couche "application" quelle qu'elle soit reste toujours indépendante de celle du "réseau" et vice-versa.

Mais il existe d'autres raisons qui justifient la découpe en couches. Les systèmes téléinformatiques sont des systèmes complexes qui nécessitent une organisation en fonctions clairement définies et nettement séparées. Ce type d'organisation assure:

- l'indépendance de l'activité des niveaux entre eux.
- la possibilité de cacher de l'information à chaque niveau, celle-ci n'est alors visible que pour l'utilisateur propre de ce niveau.
- le séquençement des événements dans le temps, de niveau à niveau.

En résumé, cette organisation facilite la conception, la mise à point et la maintenance des systèmes.

Définissons les fonctions de ces deux entités "Application" et "Réseau".



### 1.3.1 - COUCHES DES RESEAUX

Ces couches existaient dès le début de la téléinformatique et n'ont pas été modifiées lors de l'introduction des couches supérieures. Voici leurs fonctions de base:

#### COUCHE\_3: ACHEMINEMENT (NETWORK)

La couche réseau a pour rôle de fournir les moyens fonctionnels nécessaires à l'échange des informations données par la couche transport. C'est un service de bout en bout qui est responsable de l'acheminement des "paquets de données". Pour parvenir à ce résultat, de nombreux services sont fournis par la couche réseau. Citons entre autres:

- rendre transparentes les fonctions de commutation et le "routage" pour le niveau transport.
- résoudre des problèmes de contrôle d'erreurs et de flux.
- adressage.

#### COUCHE\_2: LIAISON (DATA LINK)

La couche liaison fournit les fonctions et procédures permettant d'établir, maintenir et libérer les connexions entre entités-réseau. Cette couche doit assurer l'apport, exempt d'erreurs, des blocs d'informations (sur la liaison physique). Il faut pour arriver à ce résultat des services parmi lesquels on trouve:

- contrôle de flux: l'appelant et l'appelé doivent s'arranger pour que l'information n'arrive pas trop vite au récepteur.
- séquencement: généralement cette couche doit fournir à la couche supérieure les blocs dans l'ordre où elle les a reçus.
- détection et contrôle des erreurs de transmission.

#### COUCHE\_1: PHYSIQUE\_

La couche physique fournit les caractéristiques mécanique, électrique, fonctionnelle et de procédure, permettant d'établir, maintenir et libérer les connexions entre entités DATA-LINK. Le niveau physique permet la transmission de flots binaires par une connexion, et doit assurer une compatibilité des interfaces remplissant des fonctions telles que codage, modulation du signal...

### 1.3.2 - COUCHES D'APPLICATION

#### COUCHE\_7: APPLICATION

La couche application, qui est la couche la plus externe du modèle de référence de l'OSI, permet la compréhension et l'exécution des commandes liées aux processus d'application.

L'éventail des applications de téléinformatique est très large. On y trouve des applications dont la liste qui suit n'est pas exhaustive:



- de télétraitement ou encore de ce que l'on appelle le "Remote Job Entry".
- de type bancaire.
- de type conversationnel ou interactif, allant de l'interrogation de base de données à la réservation de places, de produits, de commandes...
- de type transactionnel.
- de type "acquisition de données".
- de transfert de fichiers, de documents,...
- en temps réel.
- du domaine industriel comme télémessure, commande de processus industriel.

Ce grand nombre d'applications se caractérise par une grande variété d'une part des conventions d'échanges et de coopération, et d'autre part de caractères de contrôle et de protocoles d'affichage des données apparentes pour des terminaux de formats quelconques. Ces deux derniers problèmes seront résolus respectivement par les couches 6 et 5 dites respectivement de "présentation" et de "session".

#### COUCHE\_6: PRESENTATION\_

Cette couche a pour but de pouvoir interpréter la signification des données échangées entre les utilisateurs. Elle assure une compréhension syntaxique entre les utilisateurs, en gérant les formats des données à échanger et en effectuant les

transformations nécessaires sur les données pour les rendre compréhensibles entre matériels hétérogènes. Les fonctions que doit pouvoir effectuer le niveau sont:

- assurer la connexion et la négociation des paramètres de formatage à ce niveau.
- présenter les données échangées entre systèmes dans un format compréhensible par les applications.
- rendre les applications indépendantes des modifications d'interface ou d'application d'autres systèmes. Trois grandes catégories de protocoles correspondant à des applications différentes sont généralement admises:
  - protocole d'appareil virtuel
  - protocole de fichier virtuel
  - protocole de transfert et de manipulation des travaux.

#### COUCHE\_5: SESSION

La couche session a pour objectif de négocier les interactions entre les utilisateurs pour synchroniser les opérations effectuées sur les données. Ce niveau doit assurer:

- non seulement la connexion et la libération des entités à ce niveau, mais également le contrôle et les reprises en cas de dérapage au niveau des échanges.
- le contrôle de la chronologie et la qualité des dialogues entre processus distants de ce niveau. Cela inclus: l'activation, l'initialisation, la synchronisation (et éventuellement la resynchronisation), les échanges, le début et la fin de la session.



Ce niveau est composé d'autant de modules d'échanges qu'il y a de type d'applications (transfert de fichier ou de document, interactivité, processus...).

#### COUCHE\_4: TRANSPORT

On peut dire que c'est autour de cette couche que s'articule l'ensemble des couches supérieures orientées vers l'"application" et l'ensemble des couches "réseau".

Alors que les deux précédentes tamponnaient la couche application et de ce fait leurs programmes étaient influencés par celle-ci, la couche de transport s'en détache totalement et se rapproche plus des couches "réseau". On peut donc les considérer à peu de chose près comme standards et universelles.

La liaison entre les deux correspondants au niveau 3 (d'acheminement et de routage) étant établie, il reste à assurer l'adressage et l'identification au second niveau, celui de l'application dont les programmes d'échanges se situeront au niveau immédiatement supérieur c'est-à-dire celui de la session.

- + La fonction première de ce niveau est de contrôler l'établissement et la libération de cette connexion, ainsi que les erreurs spécifiques à ce niveau.
- + D'autres fonctions sont laissées au choix de l'utilisateur ou de la normalisation. Parmi celles-ci les plus importantes sont celles qui:

- doivent rendre l'application de plus en plus indépendante vis-à-vis du réseau.
- remplissent des fonctions de contrôle que des réseaux dits faibles ne peuvent assurer.

#### REMARQUE

Tout comme l'introduction de couches doit assurer l'indépendance entre "application" et "réseau", dans la conception de celles-ci on souhaitait également que chaque couche ne connût que sa correspondante distante, avec laquelle elle devait dialoguer "horizontalement".

Verticalement, les liaisons se feront entre les processus internes de niveaux adjacents via des interfaces similaires qui communiqueront dans les deux sens au moyen de primitives à fonctions paramétrisées. En dehors de ce genre de liaison, il y a d'autres fonctions qui rendent moins ferme l'imperméabilité entre couches.

En effet, il faut se souvenir qu'on applique la règle suivante: chaque couche négocie les paramètres pour la couche adjacente supérieure. Lorsque les paramètres sont confirmés, ils sont ensuite utilisés par cette dernière couche dans le protocole des échanges distants se situant à son niveau.



## 1.4 - PROCESSUS À COUCHES HIÉRARCHISÉES

### 1.4.1 - PRINCIPE DE LA STRUCTURATION EN COUCHES

Comme nous venons de le voir, en vue de respecter les spécifications de la normalisation des échanges ou tout simplement de se soumettre au découpage logique des programmes en modules fonctionnellement indépendants, il convient d'adopter une structuration en couches.

Pour y parvenir, il s'agira de bien répartir, sans mélange, les différentes fonctions dans les couches; chacune de celles-ci ne connaissant que les deux couches adjacentes.

- Ainsi, le logiciel d'un système faisant partie d'un réseau peut être subdivisé en unités logiques distinctes appelées couches, effectuant des fonctions spécifiques. Ces unités logiques sont en quelque sorte des machines virtuelles.
- Le logiciel est alors constitué de niveaux hiérarchiques. Chaque couche réalise des fonctions spécifiques afin de fournir des services à la couche directement supérieure, et ce en utilisant les services de la couche directement inférieure.
- Ainsi, les services du niveau (i) sont accessibles au niveau (i + 1) par l'intermédiaire d'une méthode d'accès encore appelée interface de niveau, régie par un "protocole d'intercouches" identique quelle que soit la position de cette interface de couches (fig. 1.1).

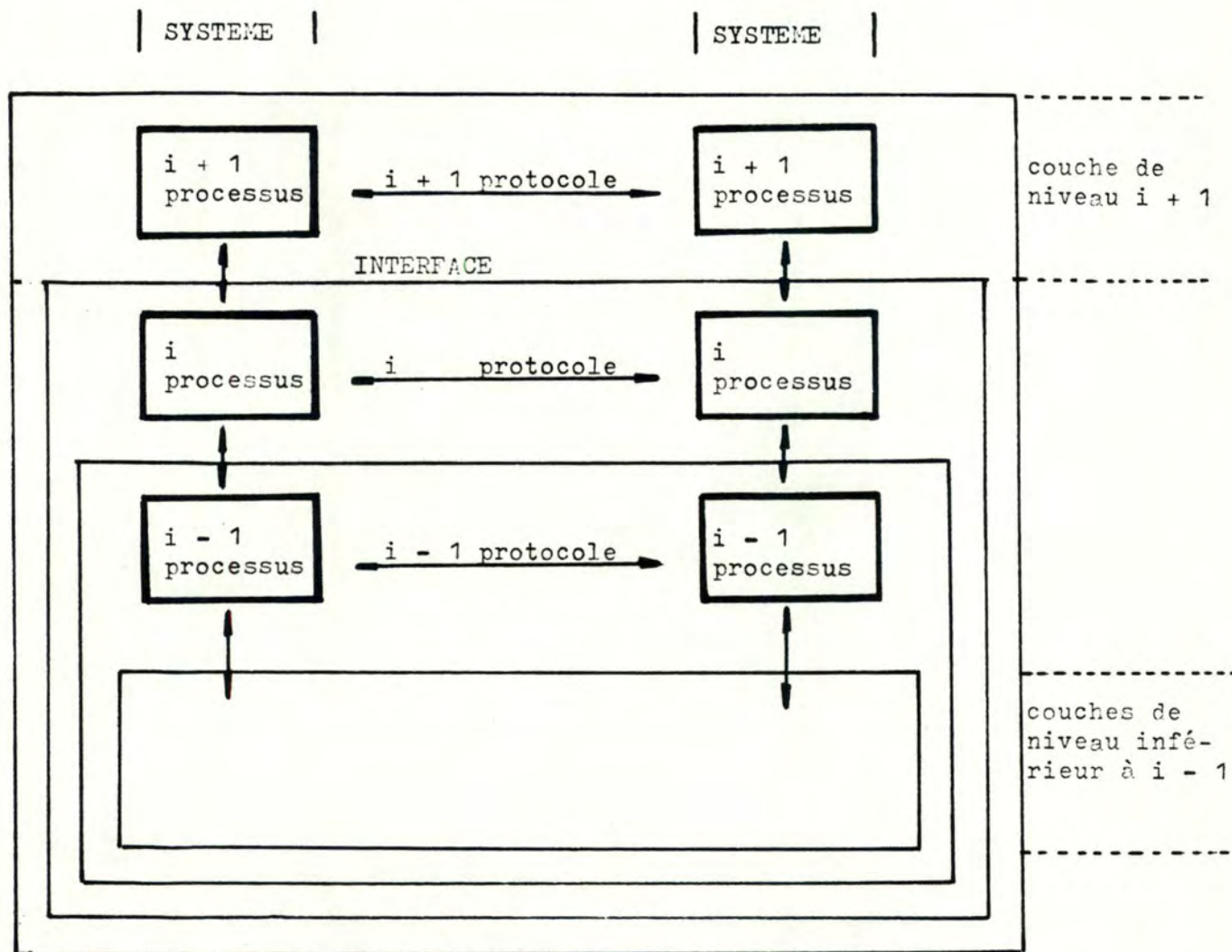


Fig 1.1

STRUCTURE HIERARCHISEE EN COUCHES



- Lorsqu'une ENTITE (composante d'un processus capable d'envoyer ou recevoir des informations) de niveau  $(i + 1)$  utilise les services du niveau  $(i)$  pour établir une connection virtuelle avec une ENTITE équivalente située dans un système distant, un point d'accès aux services  $(i)$  est réservé et identifié par la couche  $(i)$ .
- Le niveau  $(i)$  est informé de cette identification et peut par conséquent se référer à cette connection particulière.
- Plusieurs connections peuvent être établies par une même entité. Un point d'accès  $(i)$  ne peut servir qu'une seule entité  $(i + 1)$  en un instant donné.
- Une même entité de niveau  $(i)$  peut servir une même entité de niveau  $(i + 1)$  à travers plusieurs points d'accès.
- A travers l'interface de niveau sont échangés des messages de données et de contrôle. Ces derniers servent à activer les services du niveau inférieur ainsi qu'à faire rapport sur l'état d'avancement ou de fin des services. Les messages de données peuvent être:
  - soit des messages de contrôle définis par le protocole et destinés au processus distant de niveau  $(i)$ ,
  - soit des données de niveau  $(i + 1)$  délivrées au processus distant de même niveau.
- La couche  $(i)$  peut intervenir dans les deux sens. Elle peut être sollicitée:
  - par la couche  $(i - 1)$  pour réaliser une fonction destinée à la couche  $(i + 1)$ ,
  - par la couche  $(i + 1)$  pour réaliser une fonction destinée à la couche  $(i - 1)$ .



#### 1.4.2 - FONCTIONS GENERALES DES COUCHES

Comme nous l'avons vu au paragraphe 1.3, la normalisation justifie le découpage en 7 couches. Les fonctions générales:

- des couches supérieures (4, 5, 6) relatives au niveau 7 de l'application sont énumérées ci-après et détaillées au chapitre V.
- du "réseau" (1, 2, 3) sont figées par le choix que l'on a fait de celui-ci.

Dans le tableau de la figure 1.2, pour le niveau L4 et dans les suivants 1.3 et 1.4 pour L5 et L6, on trouve la liste des fonctions généralement admises aux niveaux respectifs de TRANSPORT, de SESSION et de PRESENTATION.

On y trouve aussi la position à ce sujet des constructeurs d'ordinateurs actuellement sur le marché (cfr mémoire Meyer) soit DNA, SNA, CNA et DSA.

En regard également des fonctions, la citation des primitives et paramètres qui nous serviront dans l'analyse et la synthèse de logiciels de couches nécessaires au transfert de documents.

##### 1.4.2.1 - Niveau 4 de transport (voir fig. 1.2)

Cette couche est chargée d'accomplir toutes les fonctions nécessaires pour combler l'écart existant entre les services fournis par le niveau réseau et les services nécessaires au niveau de l'échange, réclamés par le niveau application (7) via les niveaux de présentation (6) et session (5).

On peut y distinguer trois groupes de fonctions.

1 - Les services de connexion et de libération d'une liaison (1) avec:

- l'adressage (2)
- l'identification (3) appelant (contrôle) et appelé (rendez-vous).

2 - Les services de transfert de données.

- normal (4) avec des services complémentaires portant sur:
  - .. le contrôle de flux (5) et le séquençement (6).
  - .. la segmentation (7).
  - .. la concatenation (8).
  - .. le multiplexage (9).
  - .. le TSDU (bloc de données au niveau 4) (10)
- de type expédition de messages (11).
- résultant d'erreurs avec:
  - .. Détection et recouvrement d'erreurs portant sur le protocole et la chronologie... (12)
  - .. Purge (13)
  - .. Qualité de service et notification (14)

3 - Les services fondamentaux qui théoriquement doivent assurer l'indépendance des couches supérieures vis à vis du niveau (15).

REMARQUES

- a - Il y a lieu de noter que les services offerts dépendent de la classe du terminal concerné (16).
- b - Etant donné l'importance de cette couche, parfois la seule adjointe à la couche "application", nous préciserons ces fonctions au paragraphe 1.5.



# COUCHE TRANSPORT

	Fonctions ISO	DNA	SNA	DSA	CNA	P ou P	P document
1	Appel et libération d'une liaison	SCL/NSL	PC	TL	TS	P	TCR/TCA TCC/TCCA
2	Adressage	TL	PC/TC	TL	TS	p	
3	identification	SCL		TL		p	
4	transfert normal de données					P	TDT
5	Contrôle de flux	NSL	PC	TL	TS	p(u)	
6	Protocole de bout en bout	NSL	PC ITC	TL	TS	P	
7	Segmentation	NSL	PC	TL	TS	p	
8	Concatenation					p	
9	Multiplexage	NSL	PC	TL	TS	p	
10	T S D U	NSL	PC	TL	TS	p	
11	Expédition de message	NSL				P	
12	Détection et recouvrement d'erreurs	NSL	PC	TL	TS	P	TBR
13	Purge					P	
14	Qualité de service Notification		PC	TL	TS	p	
15	Indépendance du réseau						
16	Classe de service		PC	TL	TS		

SCL: Session control layer

TC : Path control layer

TS : Transnet service

NSL: Network service layer

TC : Transmission control

P/p: Primitives/paramètres

P/document: primitives pour transfert de documents.

Fig. 1.2

#### 1.4.2.2 - Niveau 5 de session (voir fig. 1.3)

La session de contrôle fournit, d'une manière générale, les services qui permettent aux processus appelant et appelé de dialoguer.

On peut en distinguer grossièrement deux groupes:

- 1 - Les services qui assurent au moyen de primitives (commandes - réponses) la connexion et la libération d'une liaison (1) comprenant:
  - les négociations
  - les tests de rendez-vous aux deux bords.

Ils portent sur:

- l'identification des adresses et des modules (2).
- la nature de la gestion des dialogues (3) (TWS, TWA, OW).

Ces derniers seront définis par les paramètres associés à ces primitives.

- 2 - Les services disponibles durant la phase de transfert des informations, implémentés selon le protocole choisi pour le dialogue.

Ils sont, comme les précédents, commandés par des primitives, notamment celles relatives à des transferts d'information de type:



- normal (4)
- message de service ou télégraphique (5)
- interactif (6)
- résultant de la mise en oeuvre d'un mécanisme de recouvrement d'erreurs faisant suite à une détection d'erreurs (8).

Ces primitives sont porteuses, tant en commande qu'en réponse, de paramètres dont certains sont négociés au préalable et d'autres sont destinés à gérer les échanges. Citons par exemple:

- contrôle de flux et de séquençement au niveau du document (10).
- inversion du sens de transmission (7).
- Largeur de la segmentation (9).



# COUCHE SESSION

	Fonction ISO	DNA	SNA	DSA	CNA	P ou p	P document
1	Connexion session (Etabl.-libération)	SCL	TC	SC	DS	P	CSS/RSS <sup>+</sup> CSE/RSEP
2	Rendez-vous (identification) Adresse au niveau session	SCL	NAU SL	SC	DS	p	
3	Gestion des dialogues	NSL TWS	DFC	SC	DS	p	
4	Echange normal de données et acc. de réception	SCL/NSL	TC DFC	SC SC	DS	P	CSUI/ RSUI
5	Message de service ou télégraphique	SCL/NSL	DFC/TC TC	SC SC	*	P	
6	Echange conversationnel	SCL	DFC	SC	DS	P	
7	Inversion transfert	NSL				P	CSCC/ RSCCP
8	Détection d'erreurs et recouvrement	NSL	CPM	SC	*	P	CSA/ RSAP
9	Segmentation	NSL	DFC	SC	*	p	
10	Contrôle de flux et séquencement	NSL	DFC/CPM	SC	*	p	
11	Multiplexage	NSL		SC	*		

SCL: Session Control layer

TC : Transmission control

DFC: Data flow control

SC : Session control

NSL: Network Service layer

NAU SC:NAU Service layer

CPM: Connection point manager

DS : Dialoge services

\* : supposée existée, mais pas d'informations disponibles.

Fig. 1.3

#### 1.4.2.3 - Niveau 6 de présentation

(voir fig. 1.4)

Les services fournis par la couche présentation permettent, après négociation entre les correspondants portant sur leurs possibilités de formatage des informations, de transmettre des données sous une forme standard comme des données enregistrées ou programmées de façon différente pour le processus d'application.

On peut diviser ces services en quatre groupes:

- 1 - Les services inhérents au "terminal" lui-même et de la classe dans laquelle il est catalogué. La mise en page est évidemment différente selon que le formatage et l'adressage portent sur:
  - a - une ligne de caractères ou le caractère considéré isolément.
  - b - la possibilité d'une tabulation de rangées ou de colonnes, c'est-à-dire horizontale ou verticale.
  - c - des champs de données liés entre eux de façons diverses. Ex: les terminaux graphiques.

Ces services comporteront des primitives (commandes/réponses):

- de connexion et de libération de la présentation. Les premières sont relatives à l'établissement soit au début d'une session (1a) ou lors d'une reprise (1b).



- de négociation de la présentation de l'image (2) après l'établissement de la liaison.
- de renégociation (3) de la présentation de l'image en cas de désynchronisation.

Ces opérations pourront se faire grâce aux paramètres de contrôle, de cadrage et de mise en forme associés à ces primitives. Ils fixeront ainsi la sélection de la syntaxe utilisée (4) et des caractères de contrôle des formats (5).

Comme nous le verrons plus tard, les performances demandées à la présentation dépendront du protocole adopté; les protocoles du terminal (6) et du fichier (7) virtuels étant les plus évolués.

- 2 - Les services pour lesquels la mise en page dépend d'un protocole de descriptions des "données" (formatage des données similaires à CODASYL pour les data-bases).

Ce protocole extrait des données elles-mêmes son propre formatage (8) ou sa propre transformation (9), les convertissant en une structure ou l'autre, selon les règles contenues dans ce protocole.

- 3 - Les services relatifs à la présentation attachent beaucoup d'importance à la négociation entre les correspondants, tant au niveau des protocoles de haut niveau (terminaux), qu'à des niveaux moins élevés. Ceux-ci portant par exemple sur:

- le compactage (10)
- l'encryptage (11)
- la conversion de code (12)
- le choix des caractères de  
contrôle (13).

Cette négociation de paramètres se réalise au niveau de la session et se confirmera normalement au niveau de la présentation.

4 - Les services relatifs au transfert des données, soit:

- normalement (14)
- lors d'une détection d'erreurs au  
niveau du document (15)
- lors du rejet du document (16).



## COUCHE DE PRESENTATION

	Fonction ISO	DNA	SNA	DSA	CNA	P/p	P document
1	Connexion présentation (établ. libér.)	NAL		PL	PS	P P	CDS 1a CDC 1 b CDE/RDEP
2	Négociation présentation image	NAL	NAU SL	PL	PS	P	CDLC/ RDCLP
3	Renégociation présentation image		NAU SL	PL	PS	P	
4	Sélection de syntaxe		NAU SL	PL	PS	p	
5	Commande format		NAU SL	PL	PS	p	
6	Protocole terminal virtuel	NAL	NAU SL	PL	*	P	
7	Protocole fichier virtuel	NAL	NAU SL	PL	*	P	
8	Formatage-Données	NAL	NAU SL	PL	PS	P	CDPB/ RDPB <sup>+</sup>
9	Transformation Données	NAL	NAU SL	PL	PS	P	
10	Compactage		NAU SL	PL	PS	P	
11	Encryptage		NAU SL		PS	P	
12	Conversion code					p	
13	Caractère de contrôle					p	
14	Transfert des données					P	CDUI
15	Recouvrement d'erreurs document					P	CDR/ RDGR
16	Rejet document					P	CDD/ RDPP

NAL : Network Application layer

PL : Presentation layer

NAU SL: NAU Service layer

PS : presentation service

\* : Supposée fournir des choses similaires, mais sans plus d'informations disponibles.

Fig. 1.4

## 1.5 - LE NIVEAU 4 - TRANSPORT

Entre les couches supérieures et inférieures, nous avons vu qu'il existe une couche charnière située au niveau 4 de TRANSPORT. Son rôle étant capital, nous en parlerons d'entrée dans ce paragraphe, pour poser les bases fondamentales de ce chapitre introductif.

- 1 - Une des fonctions essentielles de cette couche est la fonction d'adressage du processus de session, illustrée aux figures 1.5.1 et 1.5.2.

*quelle liaison*

Pour réaliser cette liaison au niveau 5 de la session, le niveau 4 de transport fait appel à des primitives (fig. 1.6) TCR, TCA, TCC et TBR. Celles-ci sont analogues à celles du protocole de base BSC (ENQ, ACK, NACK et EOT) dont les fonctions et la mise en oeuvre sont similaires (fig. 1.6.1).

On s'inspirera également de ce protocole pour déterminer les primitives minimales d'inter-couches soit: CON, ACC, NAC et DISC (fig. 1.6.2).

En principe, toutes les primitives de commande impliquent une réponse sous forme d'accusé de réception.

- Pour l'établissement de la connexion, il s'agira de TCR et TCA (fig. 1.7), toutes deux paramétrisées (fig. 1.5.1).
- Pour la libération, TCCA en réponse à TCC est obligatoire ou non, suivant la classe de service.

Les primitives maximales d'intercouches devraient être, selon la logique des fonctions à réaliser, celles indiquées à la fig. 1.8. En réalité, la vérité se situera entre les deux versions, comme nous le verrons plus loin au paragraphe 4.



- 2 - Une autre fonction essentielle de la couche 4 est de procéder à la segmentation d'un texte, pour accroître sa vitesse de transmission, tout en contrôlant sa validité par protocole. Cette fonction est également remplie, même s'il existe aux niveaux inférieurs un protocole plus ou moins performant. Ce sera le cas lorsqu'un protocole HDLC ou X25 (paquet) y est envisagé au niveau 3. Dans ce dernier cas, le protocole du niveau 4 joue alors le rôle de protocole de bout en bout.

Le niveau 4 exerce ainsi un "super" contrôle de flux et de séquençement, pour autant que le niveau dispose d'un protocole de validité et de séquençement.

Pour assurer la jonction entre les couches inférieures et supérieures, ce niveau n'est pas toujours suffisant. C'est notamment le cas, lorsque le réseau est multiple ou joue un rôle particulier. Nous en reparlerons au chapitre 5.1.

Une illustration de la segmentation est donnée à la figure 1.9.

La figure 1.10 exprime cette segmentation en termes ISO où:

- TSDU correspond au bloc physique de l'application (par exemple 1024 bytes s'il s'agit d'un écran).
- TIDU correspond à un segment obtenu après découpage du bloc physique en deux parties. La longueur unitaire du segment est donc ici de 512 bytes. Par la suite, on pourra alors appliquer sur celui-ci un protocole complémentaire.

- TPDU correspond au découpage du segment en une unité imposée par le réseau au niveau 3, ici le paquet.

3 - La figure 1.11 illustre, d'une façon générale, les fonctions des protocoles verticaux (inter-couche) et horizontaux (entre entités de L4).

Au bas de la figure sont énumérées les principales fonctions complémentaires exécutées par L4. On y trouve celles assurant l'indépendance des couches supérieures vis-à-vis du réseau.

Si les réseaux ont un objectif commun et des actions identiques (appel, données, libération), il existe néanmoins entre eux de grosses différences qui portent sur:

- le contrôle des messages qui portent des commandes identiques ou différentes.
- les protocoles qui existent ou pas. Même si le protocole est présent, des accusés de réception des commandes y sont obligatoires.
- les protocoles qui ont des contrôles performants ou insignifiants.

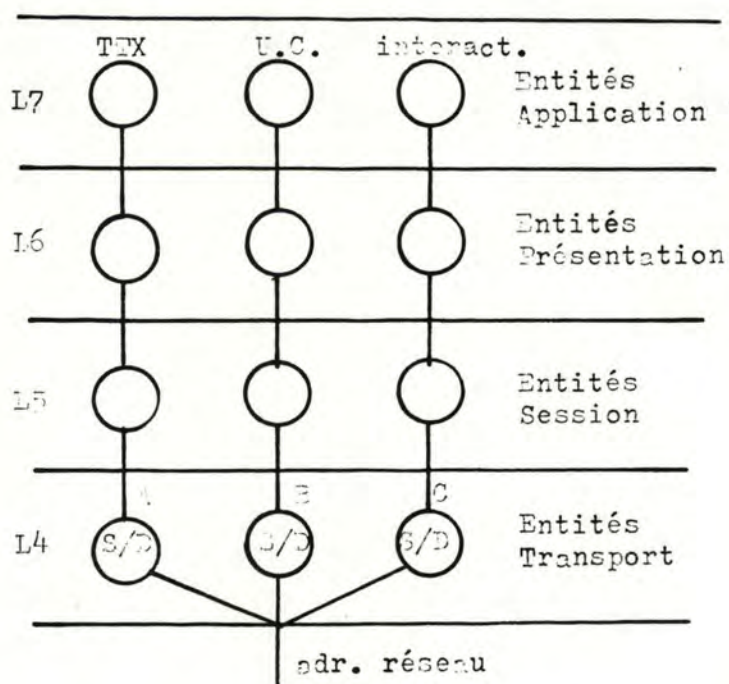
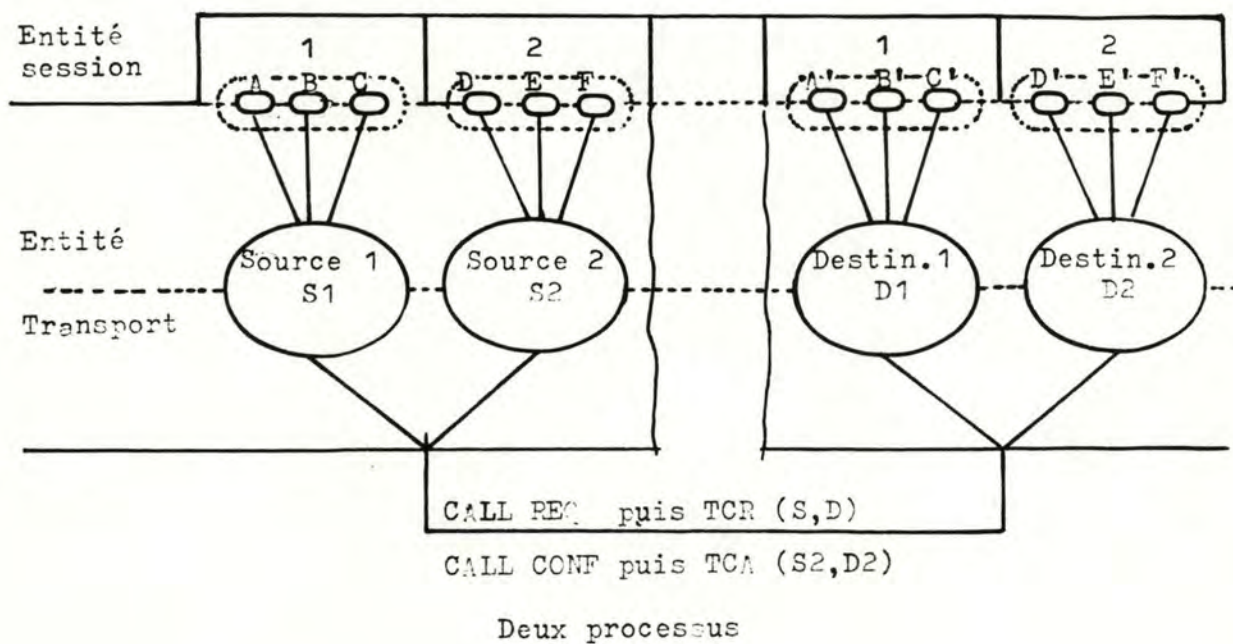
Dans ces conditions, l'indépendance vis-à-vis du réseau se résume à:

- des fonctions de L4 même si elles sont prévues ou peuvent être mises en oeuvre pour certains réseaux. Pour des réseaux à procédure non couverts par un protocole, les actions de L4 doivent impérativement être réalisées. Pour les autres réseaux à haute performance, ce n'est pas obligatoire.



- Si plusieurs réseaux doivent avoir accès aux couches supérieures, il est indispensable de procéder à une émulation des procédures et des protocoles, soit vers le haut, soit vers le bas. L'indépendance de la couche L4 vis-à-vis du réseau n'est alors possible que pour autant que l'on introduise entre L4 et L3 une couche réalisant cette émulation. Celle-ci appelée "NETWORK LAYER" est en fait officiellement admise par les instances internationales de normalisation.  
Etant également nommée "INTERFACE RESEAU", nous l'identifierons par souci de simplicité par le sigle "IR".
- Notons dès à présent que cette couche Interface Réseau est subdivisée en deux parties:
  - IR haut: adjacent au niveau 4 de transport
  - IR bas : adjacent au niveau 3.
- La communication entre ces deux sous-niveaux s'effectue via un bus. Une description plus détaillée de la composition et des différentes fonctions de IR est donnée au chapitre VI.

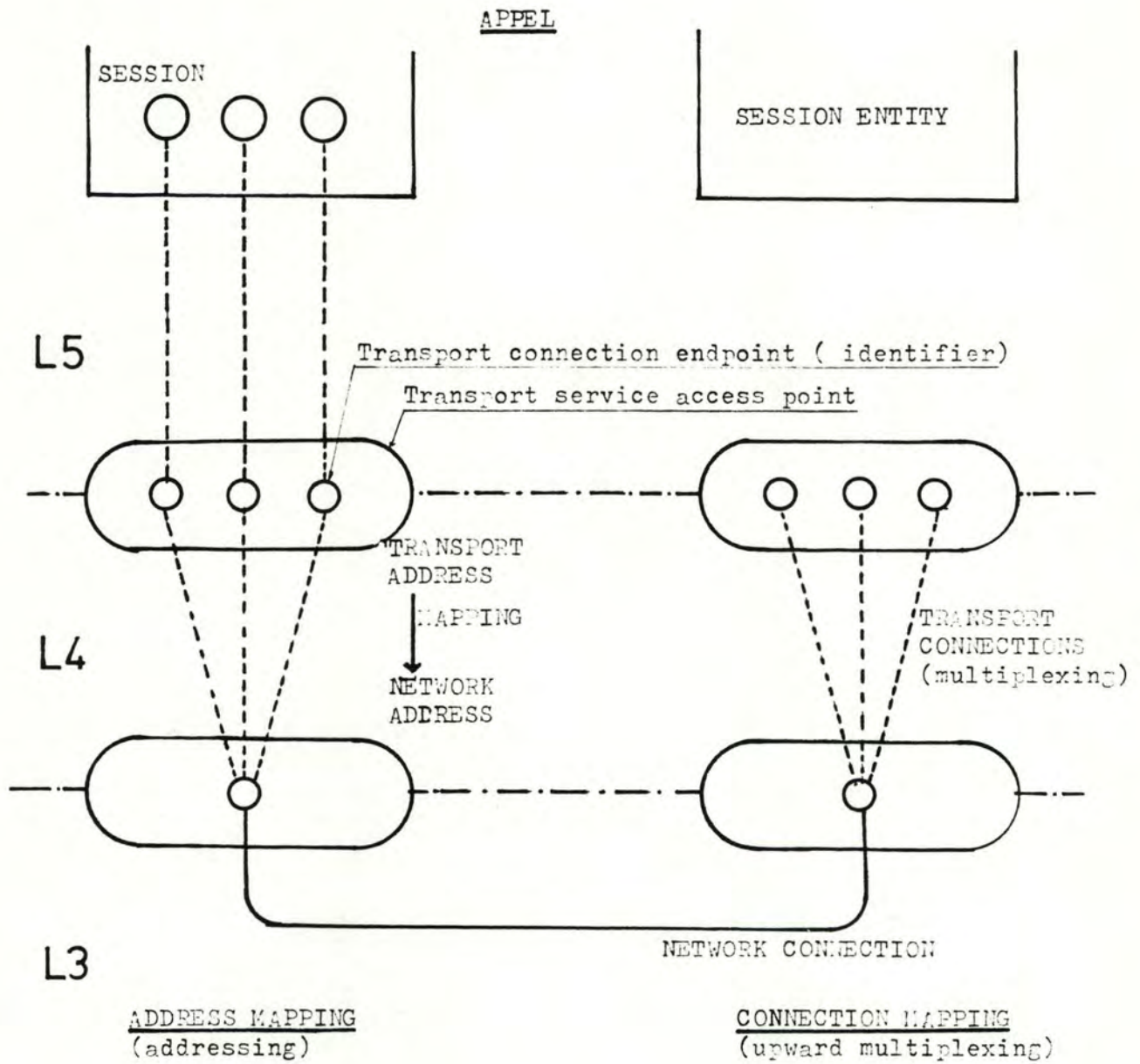
## MULTIPROCESSUS



Trois processus

- Fig. 1.5.1 -

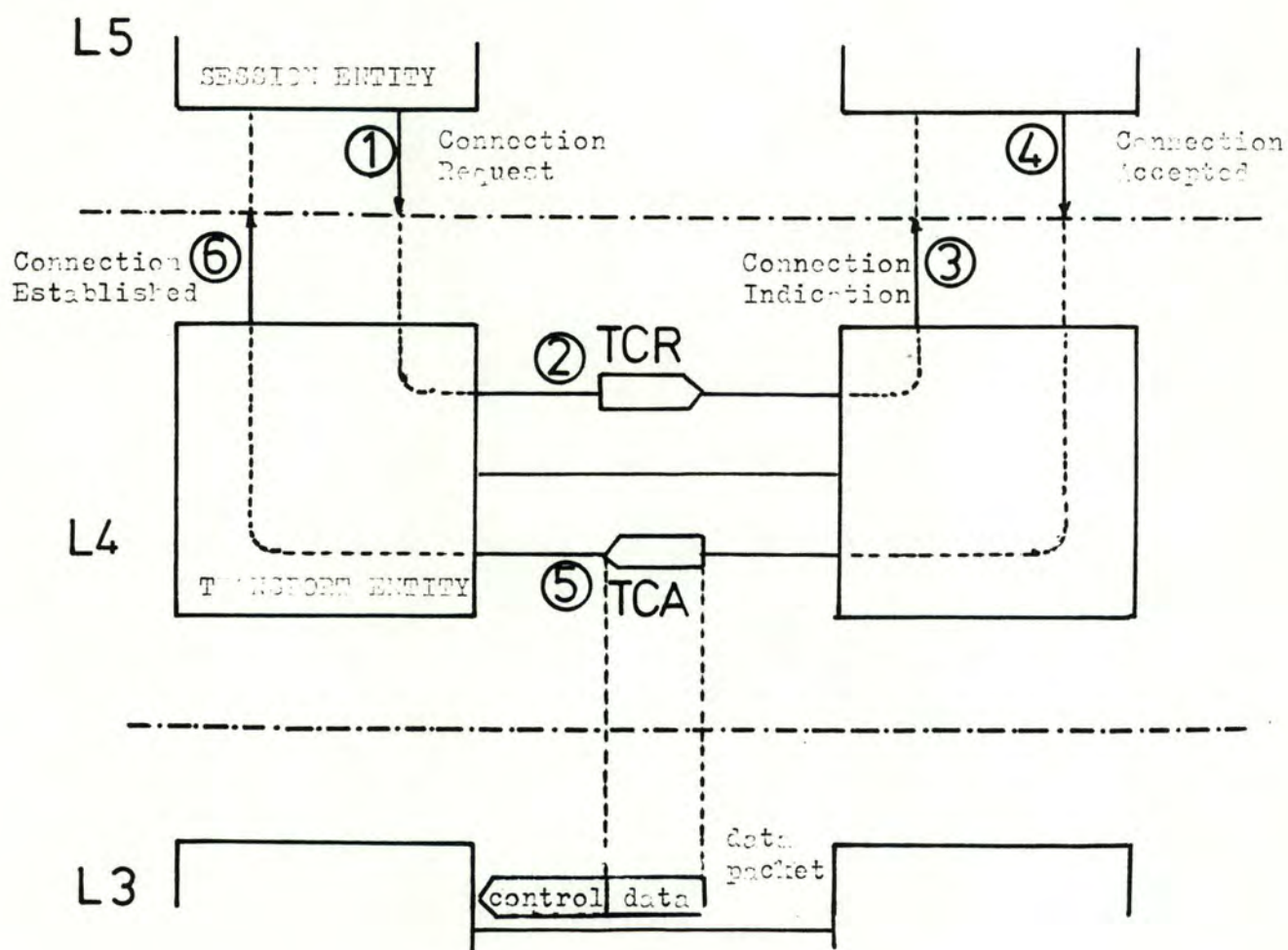




PROTOCOLE de base	PROTOCOLE entre couches L4 à L7		PROTOCOLE TTx L4 entre ST. TRANSP.
	Version 1	Version 2	
ENQ	CONN	CONN	TCR
ACK	ACC	ACC	TCA
NACK	NAC	---	TBR
ACK NACK	ACC NAC	ACC	TCA TBR
EOT	DISCON	DISCON	TCC

1.6.1                      1.6.2                      1.6.3

- Fig. 1.6 PROTOCOLES. -



- Fig. 1.7 -



## TRANSPORT SERVICES

### • Transport Connection Establishment

## SERVICE PRIMITIVES

↓ connection request  
↑ connection indication  
↓ connection accepted  
↑ connection established

---

### • Data Transfer TDSU transfer sequencing flow control error notification

↓ data transfer request (TIDU)  
↑ data transfer indication (TIDU)  
↓ TSDU delivery request  
↑ TSDU delivery indication

---

### expedited data transfer

↓ expedited data transfer request  
↑ expedited data transfer indication

---

### purge

↓ purge request  
↑ purge indication

---

### • Transport Connection Termination

↓ clear request  
↑ clear indication

Niveau 7  
Application

MESSAGE

Niveau 4  
Transport

ME SS AG E

4 segments.

Niveau 3  
Paquet

ME

E

7 paquets.

Niveau 1  
Physique

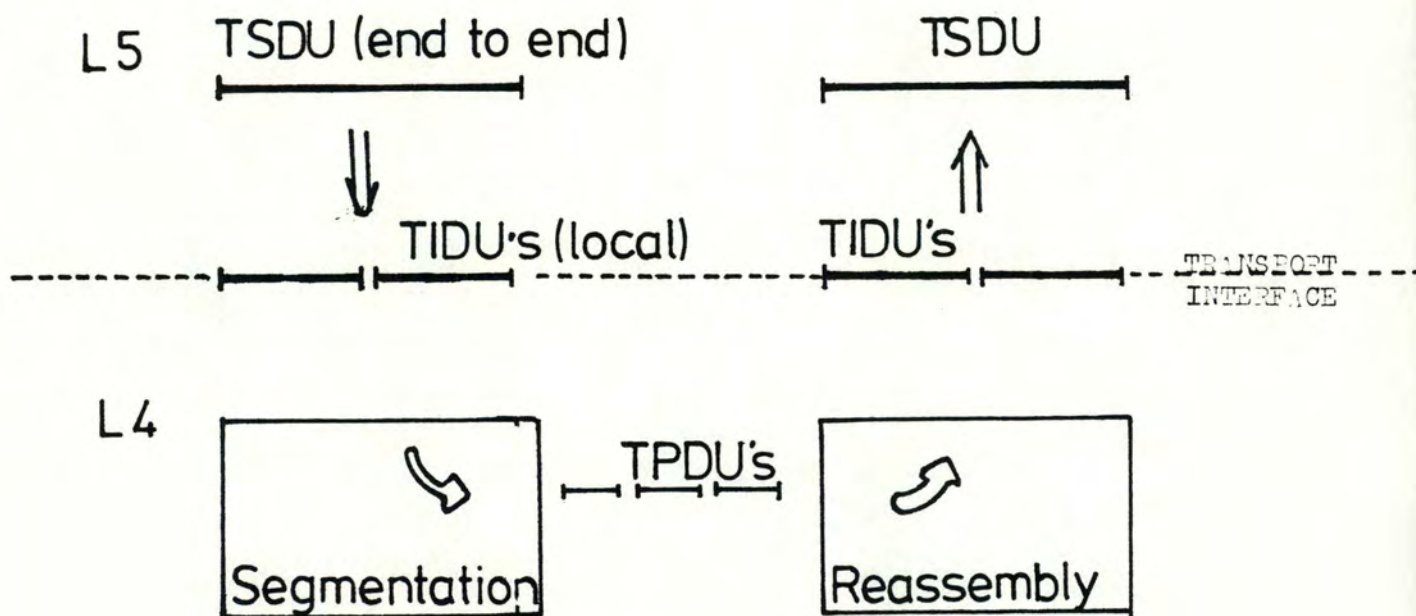
ME

E

7 trames.

Structure des unités de données



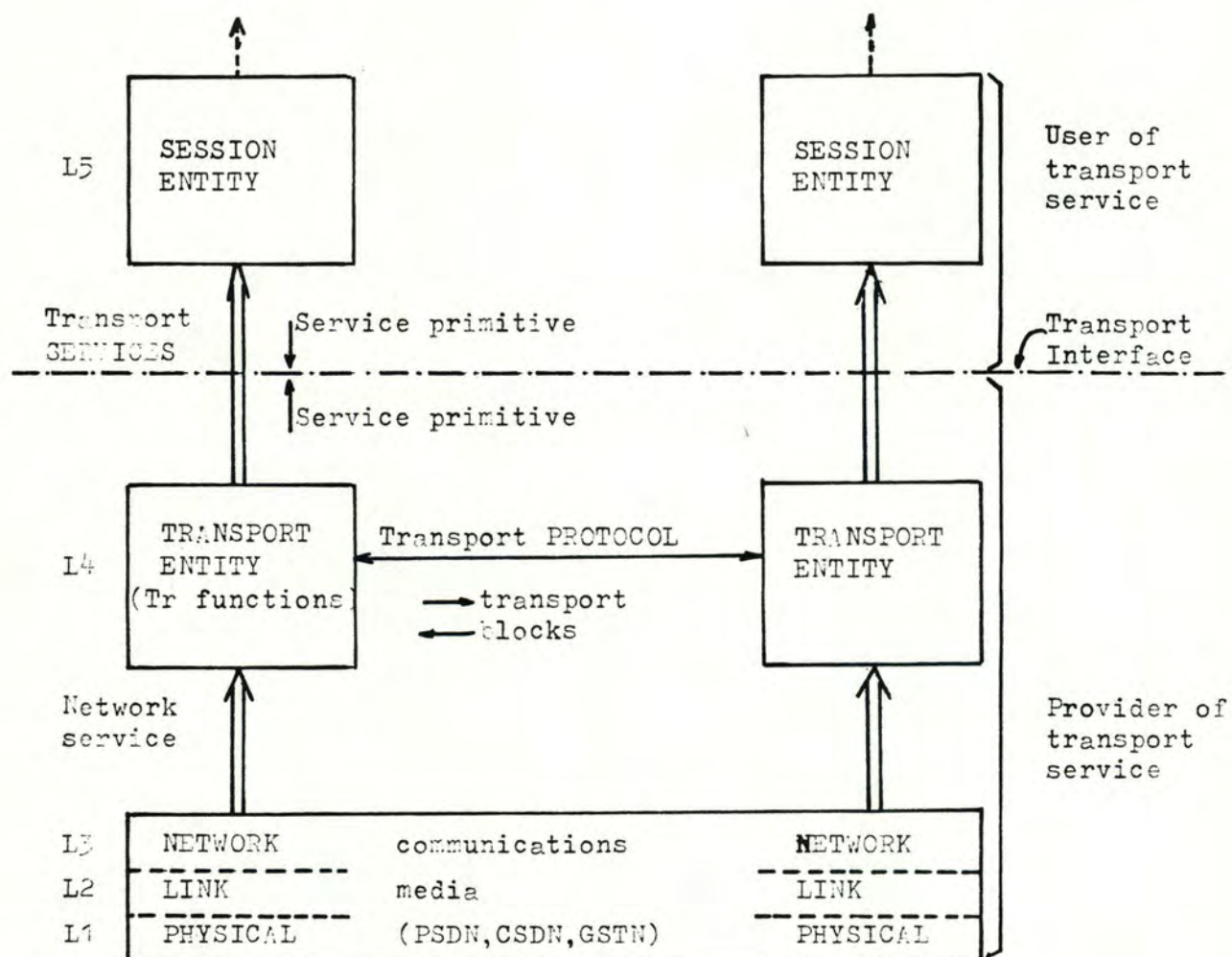


#### SEGMENTING OF TSDU'S INTO TPDUs.

TSDU : Transport Service Data Unit  
 TIDU : Transport Interface Data Unit  
 TPDU : Transport Protocol Data Unit (= Transport Data Block)

Fig. 1.10

## Network independent transport service



- Main requirements of T.S
- .Network independence
  - .End-to-end significance
  - .Error free
  - .Transparent
  - .Cost efficiency
  - .Address decoupling

- Fig. 1.11 -



## CHAPITRE II - LES PROTOCOLES DE LA COUCHE 4 ET D'INTERCOUCHES

=====

### 2.1 - PROTOCOLE BSC

### 2.2 - LES PROTOCOLES DU NIVEAU 4 ET D'INTERCOUCHES

## CHAPITRE II

=====

### LES PROTOCOLES DE LA COUCHE 4

#### ET D'INTERCOUCHES

#### 2.1 - PROTOCOLE BSC

Au chapitre précédent, nous avons vu que le protocole BSC pouvait inspirer la synthèse des protocoles de la couche 4 de transport et d'intercouches.

On peut ainsi, par assimilation, déterminer les primitives de ces deux protocoles.

Toutefois, il est bon de rappeler à partir des différentes séquences (Appel, Libération, Données, Anomalies) du protocole BSC comment les primitives de celui-ci sont mises en oeuvre et, par entraînement, celles des protocoles du niveau 4 de transport et du protocole commun à toutes les intercouches.

Aux figures 2.1 a et b, nous trouvons l'explication par l'image et les commentaires relatifs aux séquences fondamentales du BSC.



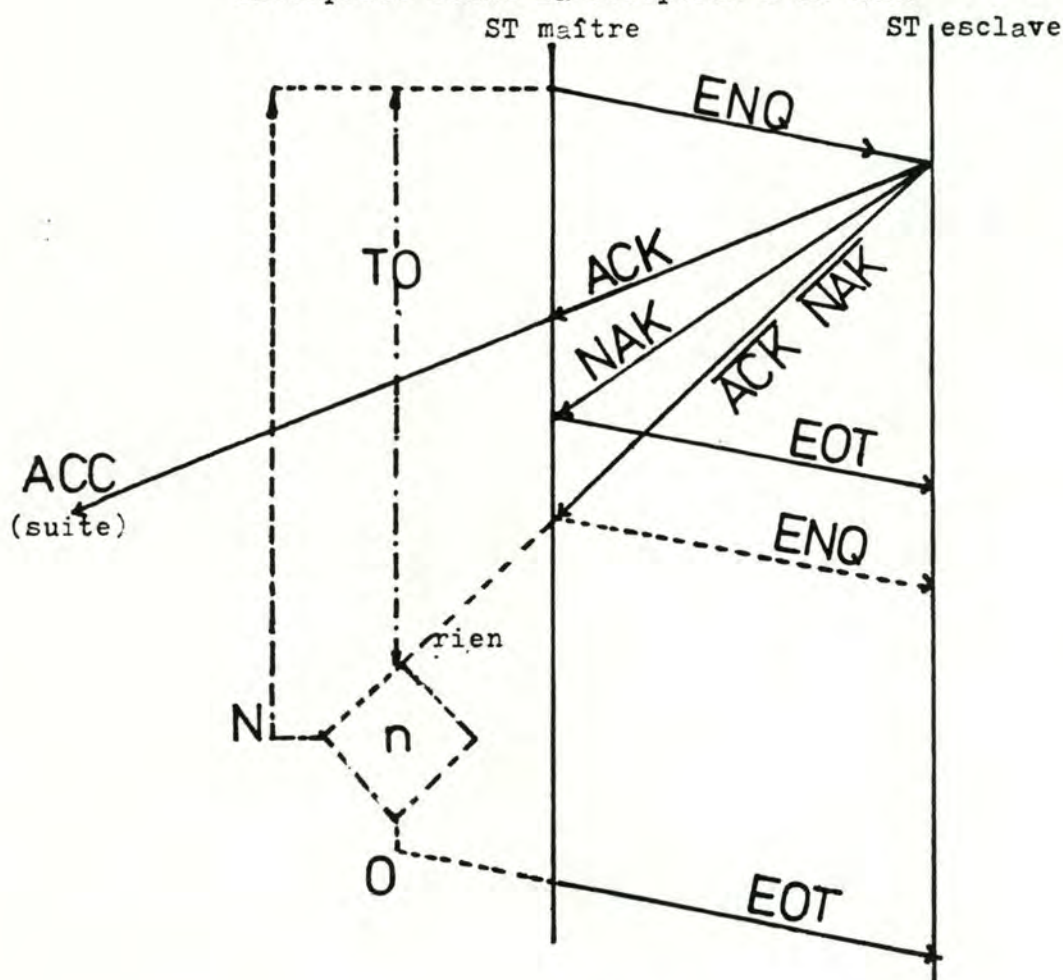
# PROTOCOLE D'APPEL / LIBERATION.

Voyons les divers cas de réponses à un ENQ :

- 1 - ACK (+): il correspond à une réponse positive.  
L'appel est "ACCEPTE", il s'en suit un enchaînement normal des blocs de données (voir figure 2.1.b).
- 2 - NAK (-): il correspond à une réponse négative.  
Il y a terminaison immédiate avec l'envoi de EOT.
- 3 -  $\overline{\text{ACK}}$ ,  $\overline{\text{NAK}}$  : il s'agit probablement d'un ACK/NAK pollué.  
Il y a renvoi d'un ENQ par la station maître (ST maître).
- 4 - Après un certain temps (time out : T.O.), si aucune réponse n'est arrivée, deux cas seront à envisager:
  - si le nombre de ENQ est inférieur à une valeur fixée (n), il y aura envoi d'une nouvelle demande (ENQ).
  - si le nombre de ENQ est égal à n, un EOT est envoyé pour indiquer la terminaison.

## Remarque:

Il existe une version simplifiée, où seul l'ACK est attendu. A l'expiration du T.O., si on n'a pas reçu d'ACK, cela est interprété comme la réception d'un NAK.

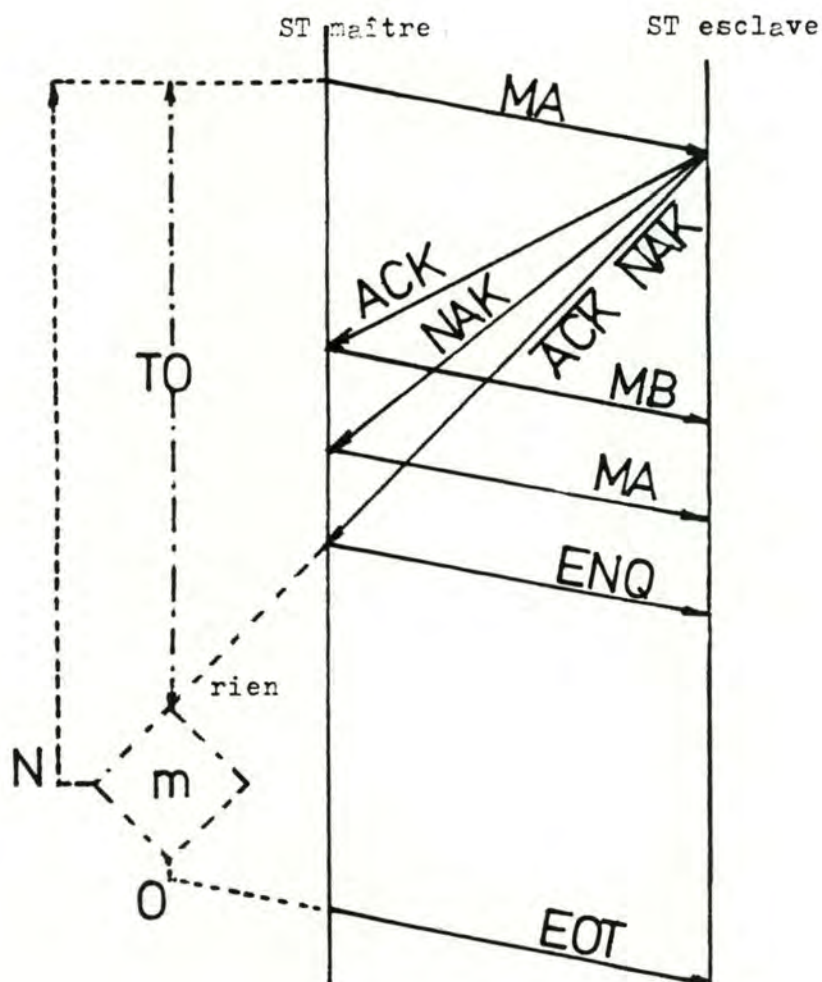


- Fig. 2.1.a -

# PROTOCOLE DE TRANSFERT DE DONNEES.

Voyons les divers cas de réponses à un transfert de message:

- 1 - ACK (+). Le bloc précédent a bien été reçu. Il y a enchaînement immédiat par l'envoi d'un nouveau bloc de données (MB...).
- 2 - NAK (-). Le bloc précédent a été jugé fautif par l'appelé. Il y aura renvoi immédiat du bloc de données MA.
- 3 -  $\overline{\text{ACK}}$ ,  $\overline{\text{NAK}}$ : il s'agit probablement d'un ACK/NAK pollué. Il en résulte une demande de précision par l'envoi d'un ENQ.
- 4 - A la fin du T.C., si aucune réponse n'a été reçue, deux cas sont à envisager:
  - si le nombre de répétitions d'envoi de MA est inférieur à une valeur (m), il y a renvoi du bloc de données.
  - si le nombre de répétitions égal à m, il y a envoi d'un EOT pour indiquer la terminaison.



- Fig. 2.1.b -



## 2.2 - LES PROTOCOLES DU NIVEAU 4 ET D'INTERCOUCHES

La figure 2.2 indique la correspondance des primitives du protocole de base et celles qui nous seront nécessaires pour bâtir par équivalence, notre système à couches.

	BSC	Protocole L4	Protocole intercouche
0	ENQ	TCR	CONN
1	ACK	TCA	ACC
2	NAK	TBR	REJ
3	ACK NAK ou	TCA TBR ou	ACC REJ ou
4	rien (to ENQ)	rien (to TCR)	rien (to CONN)
5	EOT	TCC	DISCON

- Fig. 2.2 -

Le schéma séquentiel de la fig. 2.3 précise la nature des échanges du protocole qui a été adopté en L4.

Examinons plus en détails quelques particularités de ce schéma. Nous avons :

- Le problème de EOT. En effet EOT ne peut-être transmis que par la station maître (STm). La libération s'opère donc pour L4 par TCC. Elle est :
  - valable pour la station maître et la station esclave (STe).
  - confirmée par TCCA.
- La réception du TBR.
 

Si TCR est non conforme en STe (séquence, validité des paramètres, ...) il y a répétition de TCR de la station maître à chaque fois qu'un TBR est reçu et ce, jusqu'à une limite  $n_a$  fixée.
- Un TPDU correspondant à une réception non valide en STm (séquence, validité des paramètres...). Il en résulte le renvoi d'un TBR de STm à STe, et ce  $n_u$  fois, afin de repérer un TCA.

- Remarquons enfin, qu'il n'y a pas de vérification de BCC, CRC..., ni de recouvrement éventuel d'erreurs.





### CHAPITRE III

=====

#### LES PRIMITIVES D'INTERCOUCHES



## CHAPITRE III

=====

### LES PRIMITIVES

#### D'INTERCOUCHES

Nous avons vu au chapitre I, et spécialement en 1.3 qu'il existait dans la structure en couches:

- des protocoles qui gèrent les échanges entre les mêmes couches des processus distants (protocoles horizontaux)
- des protocoles qui contrôlent les actions et les couches distinctes d'un même processus (protocoles verticaux).

Pour remplir ses fonctions, un protocole doit disposer de primitives paramétrisées. Nous préciserons dans ce paragraphe quelles sont les primitives d'intercouches, c'est-à-dire appartenant au protocole vertical de couche à couche.

Au paragraphe 1.5, et plus précisément à la figure 1.6.3, nous avons adopté les primitives fondamentales d'appel et de libération pointées à la figure 3.1. (soit CON, ACC, REJ et DISC) en prenant pour modèle le protocole de base BSC.

- a - On y adjoint, bien entendu, les primitives de transmission de données SENT et RECEIVE.

- b - On ajoute également le couple (LISTEN, CANCEL).

Au départ, les deux stations, par LISTEN, doivent se placer évidemment dans l'état de réception. Si une entité désire transmettre, elle provoque par CANCEL le basculement de la phase "réception" en phase "d'émission".

- c - On garde, en réserve, les primitives RESET et RESTART indispensables pour que le niveau 4 se synchronise avec le protocole de niveau 3, afin de réaliser les opérations de reprise et de réinitialisation.

A la figure 3.2, nous donnerons quelques précisions relatives à ces primitives, en vue de les utiliser et de les intégrer plus tard dans l'ensemble des protocoles de couches.



INTERFACE PRIMITIVE		
<u>Classe</u>	<u>Nom</u>	<u>Fonction</u>
Etablissement et clôture d'une association	CONNECT	Indique une requête pour l'établissement d'une association
	LISTEN (WAIT)	Prêt à recevoir CONNECT
	ACCEPT	Indique l'acceptation d'une association
	REJECT	Indique le rejet d'une association
	CANCEL	Révoque LISTEN
	DISCONNECT	Indique la clôture d'une association
Transfert des données	SENT	Le niveau supérieur donne des blocs au niveau inférieur afin de les envoyer.
	RECEIVE	Le niveau inférieur donne au niveau supérieur les blocs qu'il a reçus.
	CAN SENT	
	CAN RECEIVE	
Synchronisation	INTERRUPT	Envoie une interruption
	CONINTERRUPT	Prêt à recevoir une interruption
	DISINTERRUPT	Annule CONINTERRUPT
	RESET	Démarre à nouveau l'association
	RESTART	Réinitialisation

## INTERFACE DEFINITION

### INTERFACE L4/L3

a - Initialisation du terminal

READY / NOT READY ( $L3 \longrightarrow L4$ )

Paramètres:

b - Etablissement et clôture de l'association

LISTEN / CANCEL ( $L4 \longrightarrow L3$ )

Paramètres:

CONNECT ( $L4 \longleftrightarrow L3$ )

Remarque:  $L4 \longrightarrow L3$ : requête des niveaux supérieurs aux niveaux inférieurs

$L4 \longleftarrow L3$ : indication d'appel entrant pour permettre la réservation de ressources ainsi que pour prévenir les collisions

Paramètres:        - référence interne  
                          - si  $L4 \longrightarrow L3$ : numéro de l'appelé

REJECT ( $L3 \longrightarrow L4$ )

Paramètres: référence interne

ACCEPT ( $L3 \longrightarrow L4$ )

Paramètres: référence interne

DISCONNECT ( $L3 \longleftrightarrow L4$ )

Paramètres: référence interne



## c - Transfert de données

SENT (L4  $\longrightarrow$  L3)

Paramètres: référence interne

RECEIVE (L3  $\longrightarrow$  L4)

Paramètres: référence interne

RECEIVE READY (L3  $\longrightarrow$  L4) / RECEIVE NOT READY (L3  $\longleftarrow$  L4)

Remarque: on utilise un numéro de séquence interne pour la synchronisation et/ou retransmission.

Paramètres:

- référence interne
- numéro de séquence interne

## d - Synchronisation

RESET (L3  $\longrightarrow$  L4)

Paramètres: référence interne

RESTART (L3  $\longrightarrow$  L4)

Paramètres: référence interne

INTERFACE L5/L4

Analogue à l'interface L4/L3

## CHAPITRE IV - FONCTIONNEMENT GENERAL

---

### 4.1 - PROGRESSION INTERCOUCHES DES PRIMITIVES

4.1.1 - Progression de type A.

4.1.2 - Progression de type B.

1. Appel

2. Libération

3. Données

4. Pseudo-données

### 4.2 - LES OCTETS D'INFORMATION ET LEUR CHEMINEMENT

4.2.1 - Introduction

4.2.2 - Octets d'information en entrée (BIU/IR)  
et en sortie (BOU/7)

4.2.3 - Octets d'information à l'arrivée (BINR/7)  
et au départ (BONR/IR)

4.2.4 - Programmes de couches

4.2.5 - Mécanismes des opérations

### 4.3 - IMBRICATION DES OPÉRATIONS

4.3.1 - Séquence de base des opérations

4.3.2 - Illustrations des séquences



## CHAPITRE IV

=====

### FONCTIONNEMENT GENERAL

#### 4.1 - PROGRESSION INTERCOUCHES DES PRIMITIVES

Les primitives déterminant les actions à prendre entre les couches de chacune des stations ont été définies au chapitre 3.

Nous rappelons les plus fondamentales:

- pour l'appel et la libération: CON, ACC, REJ, DISC.
- pour le transfert de données: SENT, RECEIVE.
- pour permettre l'accès à l'une ou l'autre couche:
  - RR: signifiant "Ready Receive"
  - RNR: signifiant "Ready Not Receive".

Des paramètres tels que Adresse source, Adresse destination, fonctions complémentaires, leur sont associés

Ces primitives doivent transiter dans les couches:

- pour tester si la couche adjacente est accessible.
- pour provoquer l'opération assignée à la couche à un moment précis.

On peut diviser en deux, les types de progression des primitives dans les couches.

#### 4.1.1 - PROGRESSION DE TYPE A

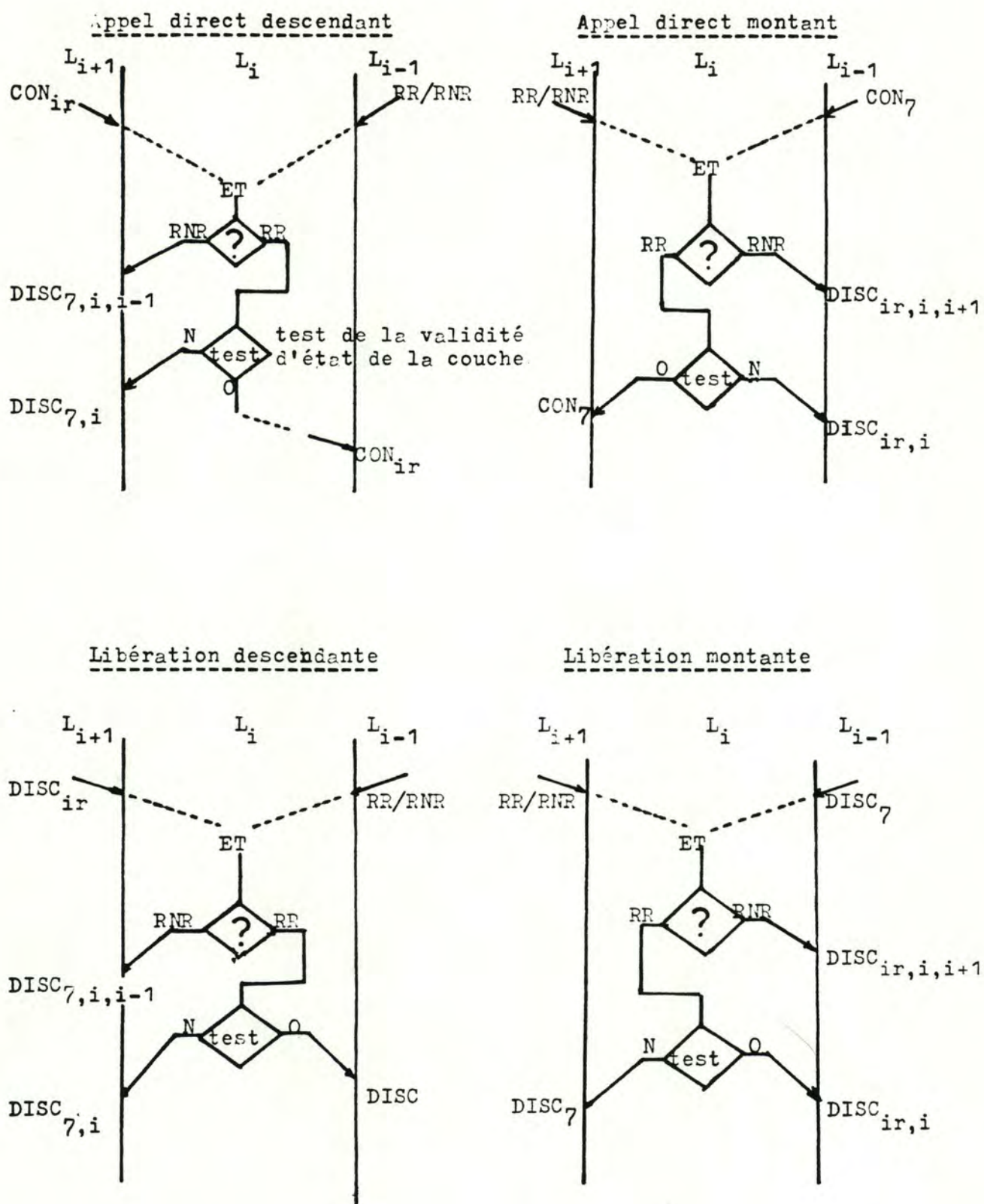
Il s'agit d'une progression de bout en bout, en transit à travers toutes les couches, de deux primitives utilisées pour la connexion (CON) et la déconnexion (DISC). Elles se manifestent pour un appel direct ou une libération au niveau du réseau pour IR (= niveau de l'intercouche réseau) de et vers le niveau 7 de l'application; c'est-à-dire:

- l'appel (ou la libération): une progression de  $L_7$  à  $L_{IR}$  de type top-down pour l'appelant.
- l'appel (ou la libération): une progression de  $L_{IR}$  à  $L_7$  de type bottom-up pour l'appelé.

#### REGLES (Fig. 4.1.1)

- 1 - La primitive descendante (CON/DISC) transite du niveau 7 au niveau IR en ne procédant qu'à des tests de validité d'états de chaque couche. Si l'état de la couche n'est pas valide cela entraîne une opération RAZ ou repos. (Processus similaire pour l'ascendant).
- 2 - Les causes de non progression sont:
  - a - le refus de la couche adjacente par RNR. Si le RNR de la couche (i-1) affecte la couche i, il y aura un DISC vers les couches supérieures. La primitive prendra comme paramètres n, i, i-1 indiquant successivement la destination, le numéro de la couche affectée et le numéro de la couche "rebelle" (refusante) (cfr. les figures 4.1.1 CON et DISC descendante). On notera l'inverse n, i, i + 1 pour les figures CON et DISC ascendantes.





- Fig. 4.1.1. -

- b - la non validité d'état de la couche qui n'est pas conforme (état présent) pour effectuer l'opération demandée (état suivant dans la table des états).

Dans ces conditions, la primitive DISC se dirigera vers la couche d'où vient la primitive de commande et sera affectée des paramètres  $n, i$  indiquant successivement la destination et le numéro de la couche en défaut.

- c - Il peut y avoir refus préalable d'une liaison par une application. Ainsi, l'opérateur ou le programme d'application peut préalablement positionner un NAK au niveau "réseau". En cas d'appel distant, celui-ci est alors rejeté. De plus, seuls les messages admis par le protocole arriveront à "l'écran" ou application, et l'opérateur ou le programme pourront fournir un ACK (traduit ACC). Les messages d'appel se butant à un NAK ou dont le contrôle de qualité par BCC/CRC est négatif seront refusés.



#### 4.1.2 - PROGRESSION DE TYPE B

Il s'agit d'une progression de primitives pas à pas jusqu'à un niveau quelconque entre L7 et LIR, dont la valeur est assignée par un paramètre au départ de cette primitive d'intercouche.

Il y a scrutation par le moniteur de la pile des zones ZL. Le moniteur décode 4 cas par comparaison entre le code d'en-tête de ZL et de la table des états des couches. Les 4 cas sont les suivants:

##### 1 - Appel (fig. 4.1.2)

Après que la liaison réseau ait été faite, il y a initialisation des couches 4, 5 et 6 par des primitives ascendantes selon le dialogue:

```
TCR —→ TCA
CSS —→ RSSP
CDS.
```

Ces dialogues ont pour origines respectives les primitives paramétrées générées par la couche "IR" selon le résultat du test de ZL par le moniteur. Il s'agit pour les commandes de ACC 4, 5, 6 (ou REJ 7) et pour les réponses de CON 4, 5, 6 (ou DISC 7).

En sortie par exemple, la primitive de départ (tjs ascendante) porte le numéro du niveau à atteindre.

Contrairement à ce qui se passe pour le type A opérant un transit de bout en bout avec simplement le test de validité des couches traversées, dans le cas du type B,

il y a un niveau "cible" à atteindre.  
Il y aura non seulement un test mais également un choix et une prise de décision de l'état suivant avec génération de la primitive de couches, adjonction des paramètres et enregistrement dans la zone ZE.

Notons cependant qu'en entrée les opérations seront inversées.

A présent, reprenons dans le détail les opérations relatives à l'APPEL et distinguons:

- 1a- (fig. 4.1.3) un appel prolongeant du côté appelé la demande de connexion faite par l'appelant à travers le réseau, soit pour
- L4: TCR
  - L5: CSS
  - L6: CDS.

Il sollicite l'intercouche via ZL pour qu'elle transmette une primitive CON (ACC) ou DISC (REJ) à destination de la couche concernée ( $n = 4, 5, 6$ ).



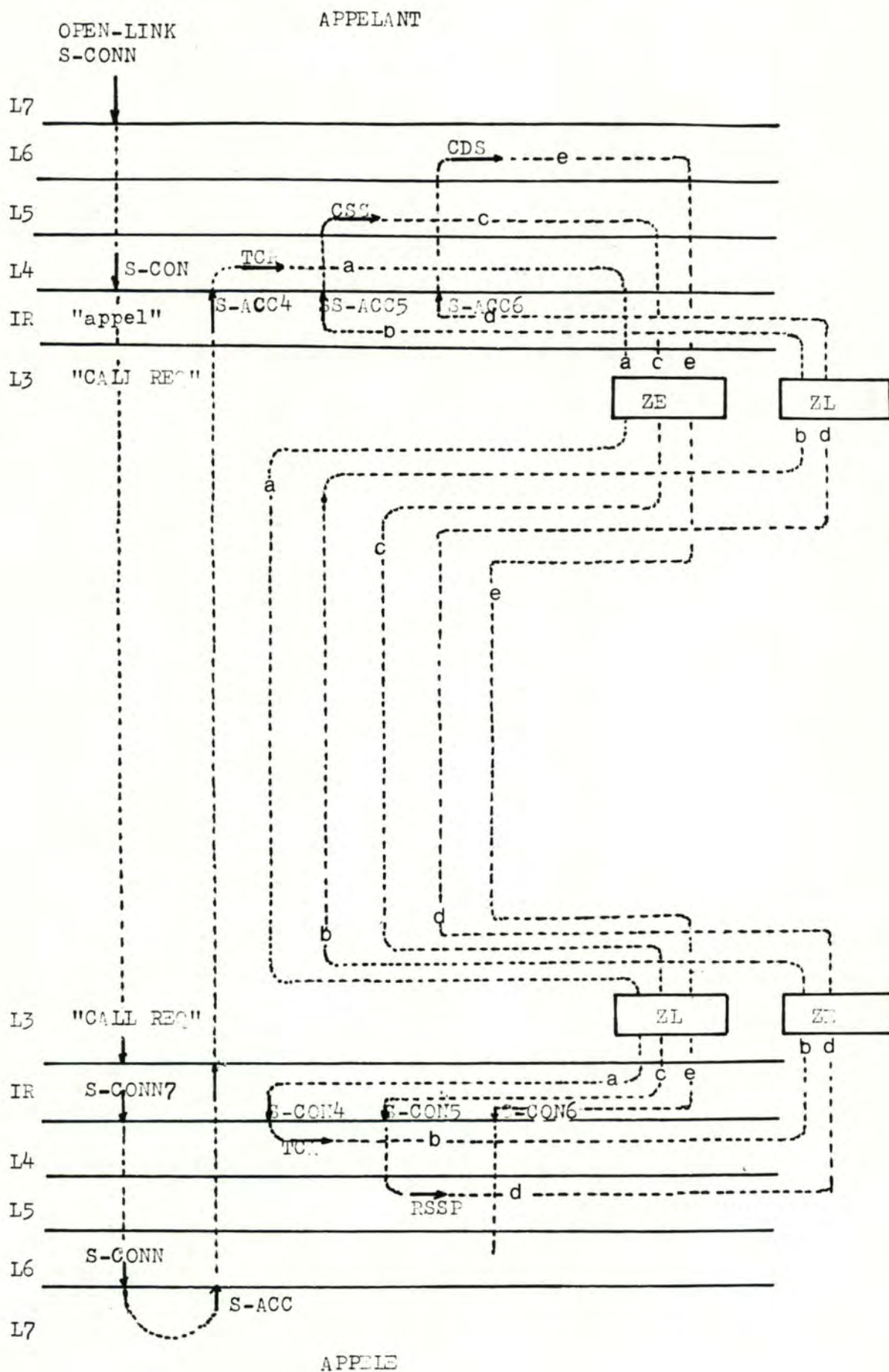
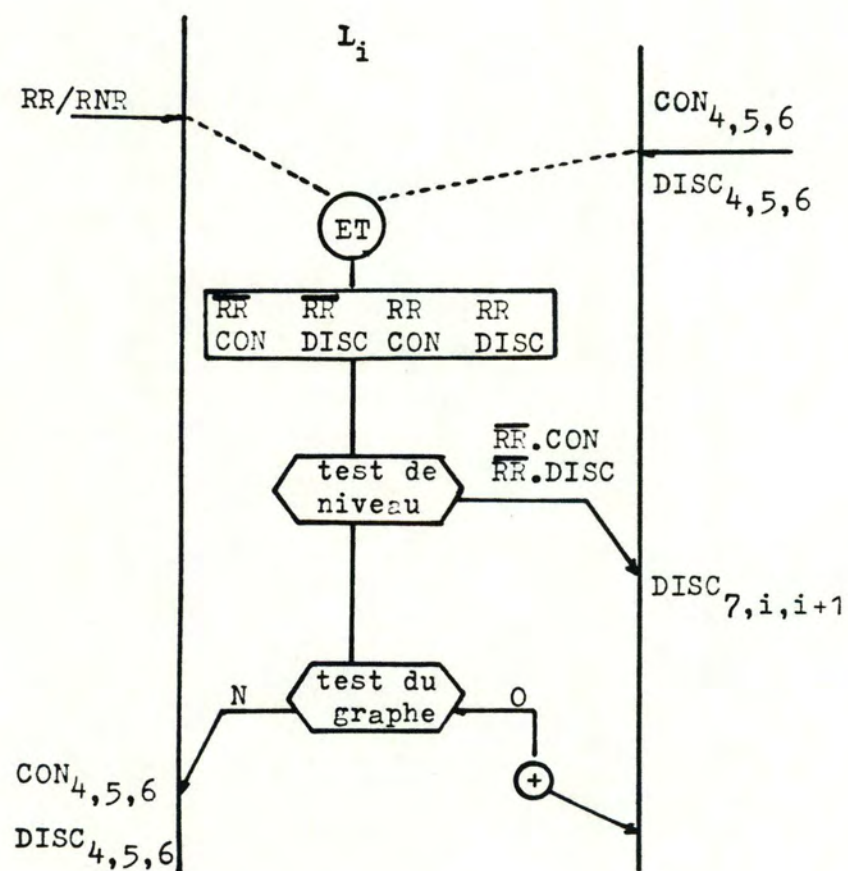


Fig. 4.1.2.



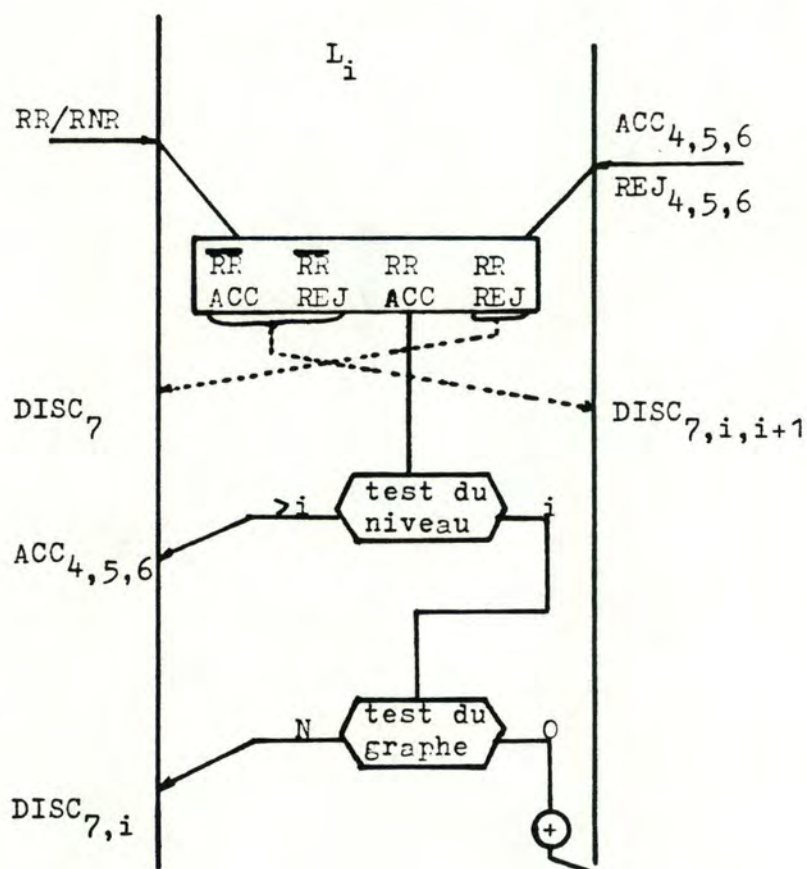
+ Réveil du programme de couche pour l'envoi de :  
 TCC ou TCA ; s'il a reçu un TCR  
 RSSP ou RSSN ; s'il a reçu un CSS

- Fig. 4.1.3 -



1b- (fig. 4.1.4): une réponse à un appel du réseau qui a été accepté ou refusé:

- Dans l'ordre chronologique, le code relevé par le moniteur dans ZL(extrait de la pile) va déterminer dans l'intercouche une primitive ACC4, ACC5, puis ACC6.
- Le cas d'un rejet du réseau donne lieu a un REJ7, DISC7 allant vers l'utilisateur indiquant à celui-ci quelles sont les causes éventuelles du rejet.



+ Réveil du programme de couche pour l'enregistrement de TCR, CSS, ou CDS avec leurs paramètres dans la zone Ze et leur transmission vers l'appelé.

- Fig 4.14 -

## 2 - Libération (fig. 4.1.5)

Alors que l'appel "réseau" s'effectue obligatoirement de l'appelant vers l'appelé, la libération peut se faire des deux façons suivantes:

2a- par l'appelant: le relâchement s'opère en deux temps:

§1 - la RAZ des couches successives avec pour chacune d'elles le dialogue des primitives de fin (CDE, CSE...) et des réponses (RDEP, RSEP...). Si cette opération se fait normalement, on passe en §2.

§2 - L'envoi de la primitive fondamentale DISC aboutissant à l'appelé.

2b- par l'appelé: le relâchement par l'appelé ne peut se faire que si l'appelant a marqué son accord au préalable; d'abord au niveau de la couche "Network" et ensuite au niveau 7

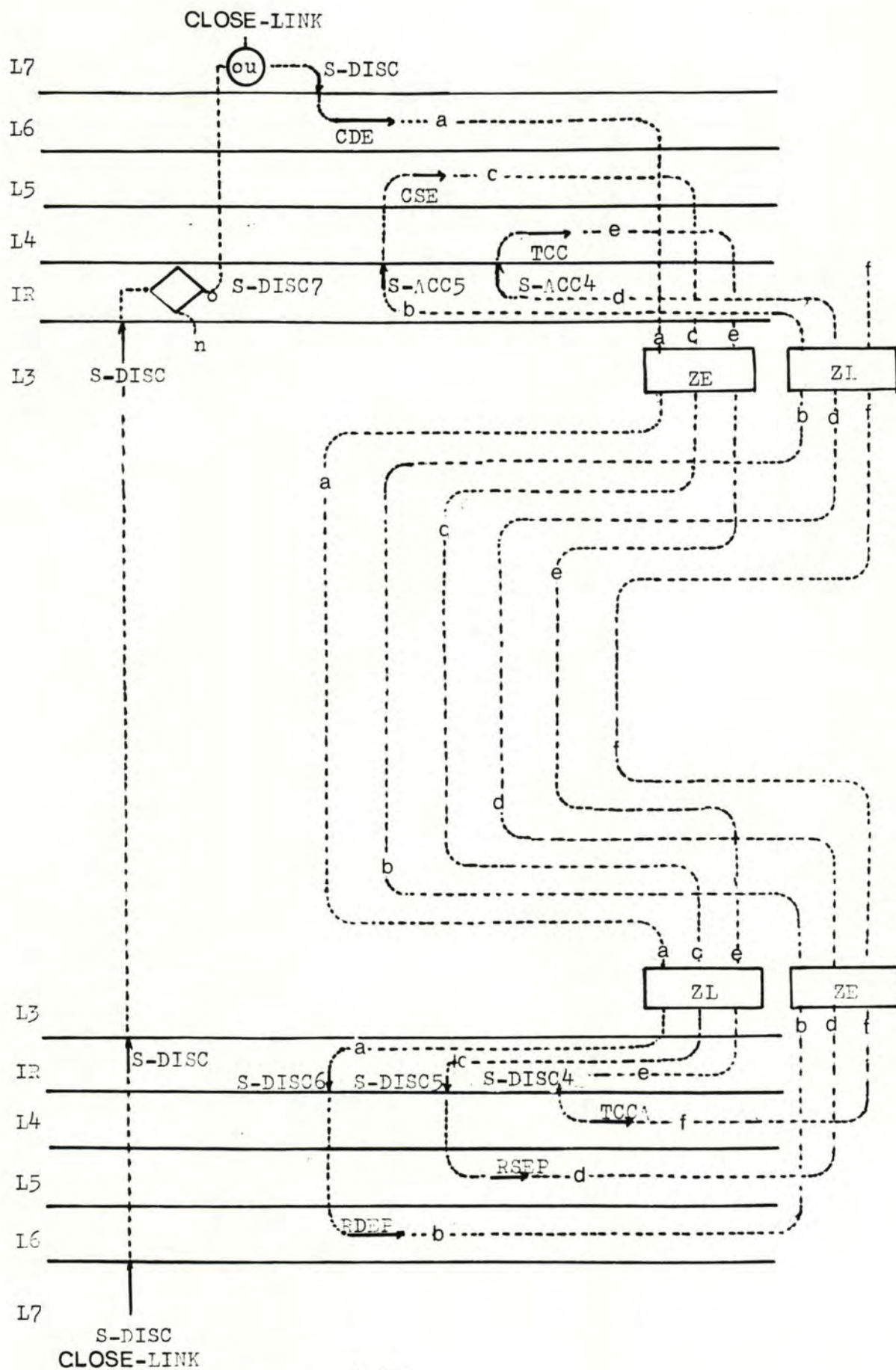
Cette demande de libération de l'appelé parvient à l'appelant (écran-application). Si ce dernier accepte cette requête, cela provoque la libération, comme si elle émanait directement de l'appelant (cas 2.a).

## 3 - Données (fig. 4.1.6)

Il s'agit dans ce cas de données pures ou d'informations, émises ou reçues, par l'utilisateur ou le programme d'application.



## APPELANT



## APPELE

Fig 4.15.

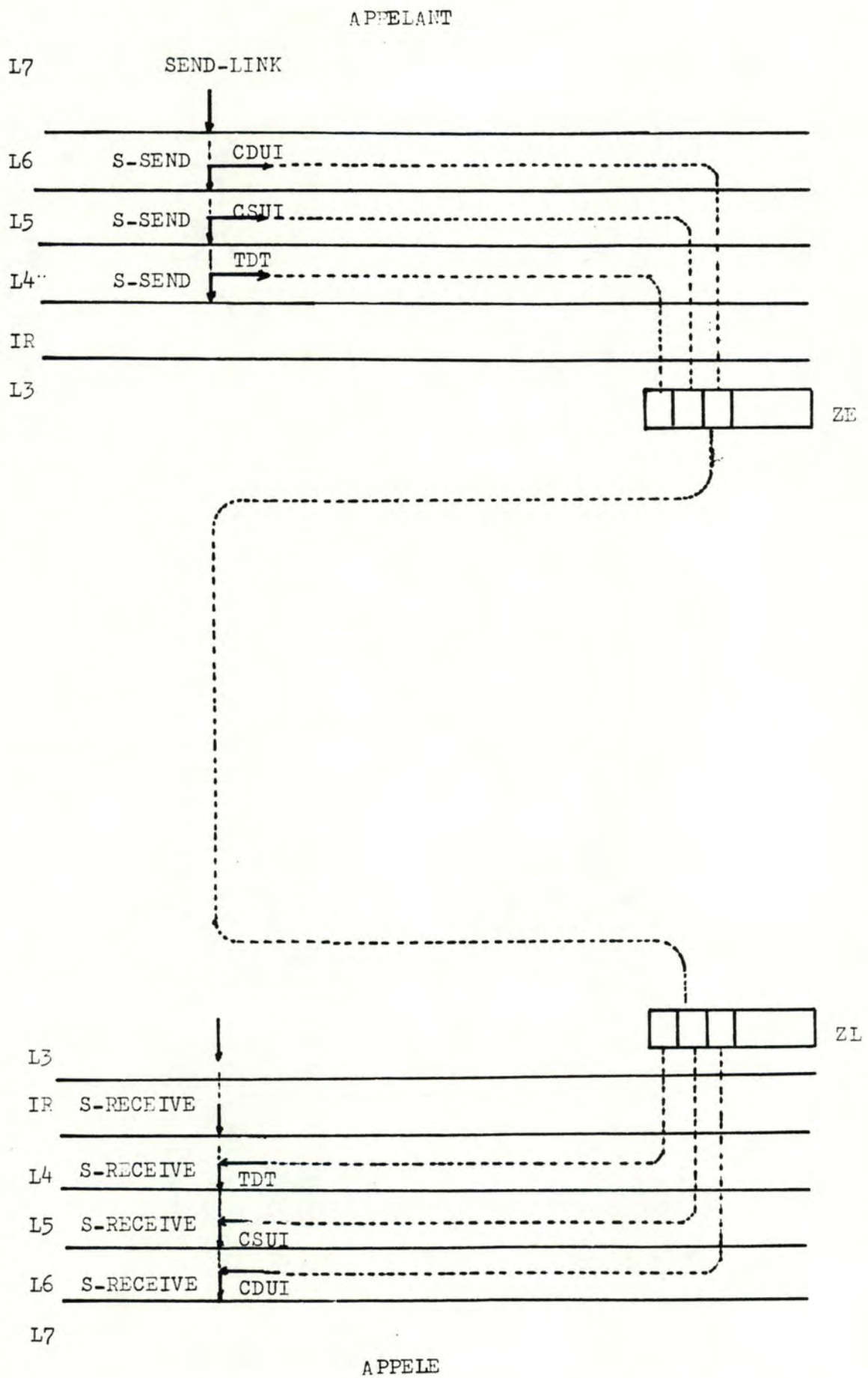


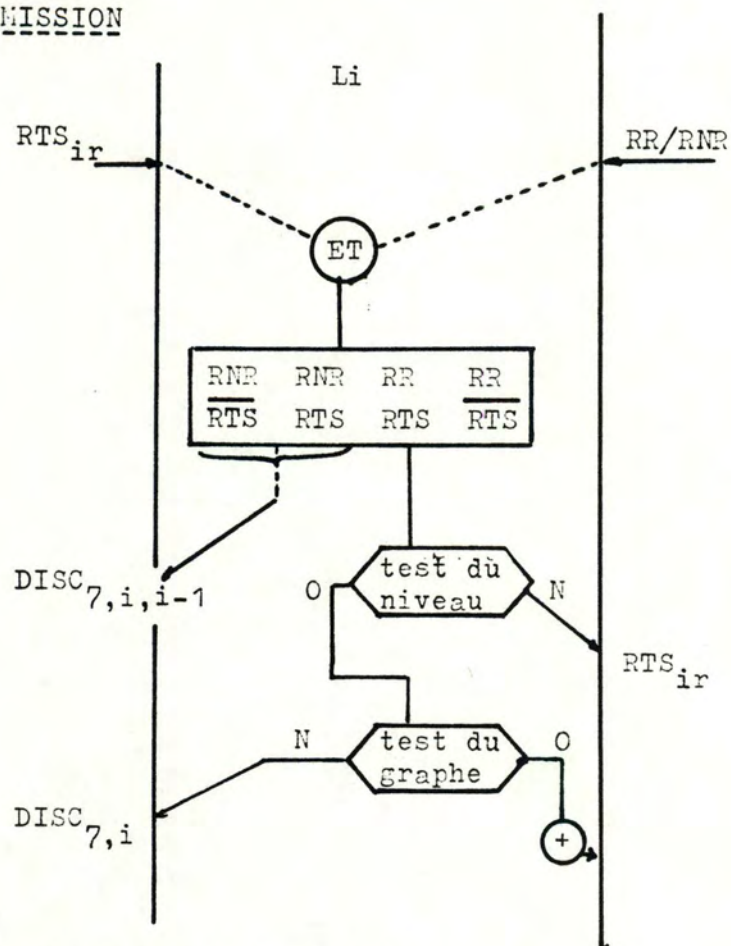
Fig 4.1.6.



Les primitives RTS (S. SEND) pour l'émission et RTR (S. RECEIVE) pour la réception sont respectivement descendantes et ascendantes.

Pour chaque couche, il y a après le test de validité (fig. 4.1.7) :

- pour RTS: le choix de l'état suivant avec enregistrement des primitives et des paramètres dans ZE à savoir: CDUI au niveau 6, CSUI au niveau 5 et TDT et ses paramètres au niveau 4. Les paramètres de TDT sont: les paramètres de fin de texte et les T (R), T (S) - incorporés après consultation du module - qui permettent le contrôle de séquençement des segments.
- pour RTR: la validité résulte de la comparaison de l'état du graphe et du code lu dans ZL par le moniteur pour chacune des couches. Le contrôle au niveau 4 relatif à TDT portera particulièrement sur T (R) et T (S).

EMISSION

+ Contrôle de N(R), N(S)  
EOT.

Remplissage de Zc par les primitives et leurs paramètres.

- Fig. 4.1.7 -



#### 4 - Pseudo-données

Il s'agit de primitives "hors séquence normale" des couches 4, 5 et 6, dialoguant à ces niveaux et où l'accès à ceux-ci peut être direct.

Elles sont traitées de façon similaire aux données et s'y intègrent même pour les différentes couches. C'est la raison pour laquelle nous les avons désignées par le nom de "pseudo données".

Lors du dialogue entre couches, un certain nombre de primitives n'apparaissent pas dans les échanges d'appel , de libération ou de données.

Elle seront traitées du point de vue primitives d'interface de données (RTS, RTR) puisqu'elles se manifestent durant les échanges de données. A la différence de ces dernières, elles n'auront pas le niveau 7 de l'utilisateur comme origine ou destination, mais bien un niveau intermédiaire.

Les primitives précitées sont:

	Commande	Réponse
niveau 4	S-TBR	R-TBR
niveau 5	CSCC	RSCCP
	CSA	RSAP
niveau 6	CDCL	RDCLP
	CDPB	RDPBP
	CDE	RDEP
	CDD/CDR	RDDP/RDRP
	CDC	

A la réception: - du côté appelé, les  
commandes de l'appelant  
- du côté appelant, les  
réponses de l'appelé.

A l'émission : on retrouve l'inverse.

Les deux opérations étant similaires, on  
va donc considérer le cas de la réception.

Premièrement, il y a lieu de vérifier si  
le moniteur peut

- par comparaison des codes des primitives  
précitées,
- grâce aux renseignements fournis par  
les tables d'état (numéro couche/état  
présent),

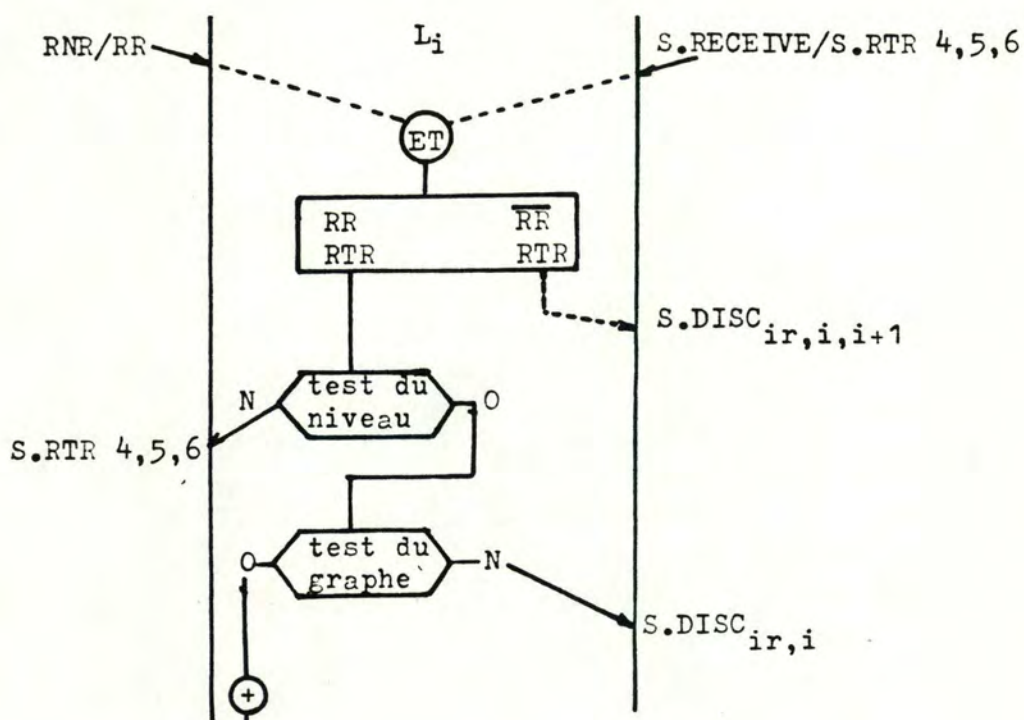
donner un numéro de couche "cible" à la  
primitive d'intercouche.

En d'autres termes, après l'analyse de ZL  
(fig. 4.1.8)

§ 1. le moniteur, grâce au code, essaye  
de fournir le numéro de la couche  
à atteindre.

§ 2. la couche qui reçoit un RTR 4,5,6  
essaye de situer celui-ci dans le  
graphe afin d'en déterminer la  
fonction.





- + Traitement du cas:
- . génération de primitives
  - .. remplissage de Ze par les paramètres
  - . fin

Fig. 4.1.8

Quand l'opération est terminée, il n'y a pas de retour au moniteur à travers les couches car, étant donné que le module du programme de couche est terminé, il y aura saut vers le moniteur à partir de FIN du programme de couche.

Ce retour au moniteur est indispensable pour que celui-ci puisse communiquer ZE à la couche réseau.

## 4.2 - LES OCTETS D'INFORMATION ET LEUR CHEMINEMENT

### 4.2.1 - INTRODUCTION

La compatibilité totale des couches supérieures avec les couches de n'importe quel réseau est possible grâce à l'introduction d'une couche intermédiaire "Network Layer" ou "Interface Réseau" (I.R.) (fig. 4.2.1) dont les fonctions seront:

- 1 - De ramener les commandes et réponses émanant (ou à destination) des différents réseaux vers (ou en provenance) des couches supérieures en un ensemble de commandes et réponses standardisées.

Ces fonctions correspondent aux primitives adoptées et se réduisent l'une comme l'autre au positionnement d'un bit dans des octets dits "d'information" destinés aux couches supérieures de l'application.

Ces octets sont au nombre de 4 (BOU/BONR et BIU/BINR) selon d'une part, leur origine et leur destination (Niveau 7 ou IR) et d'autre part, le sens des fonctions qu'ils opèrent soit:

- couches inférieures (ou Réseau) vers les couches supérieures (ou Application)
- couches supérieures vers couches inférieures.

La localisation de ces 4 octets et la signification de leurs bits est illustrée à la figure 4.2.2 a/b. La justification de leur contenu sera donnée au chapitre 4.2.2.



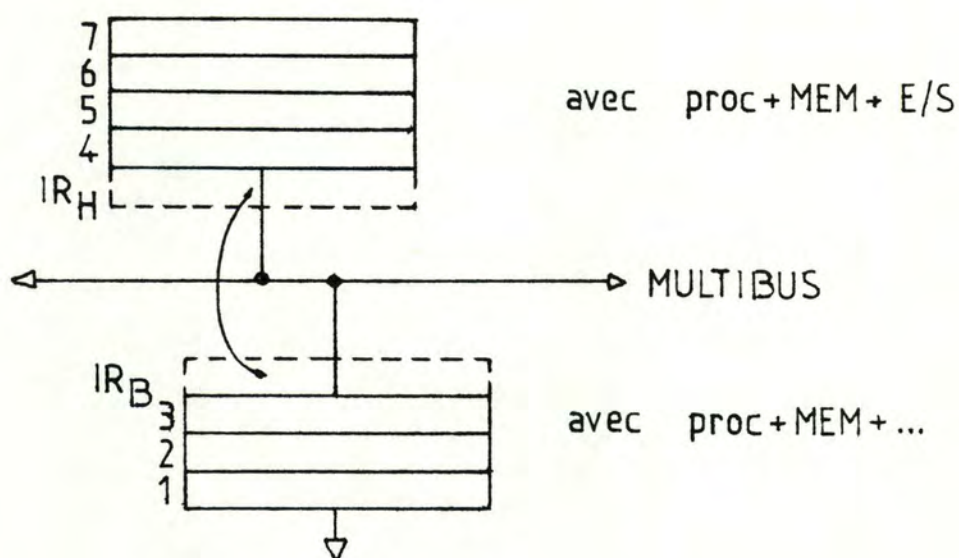


Fig 4.2.1.

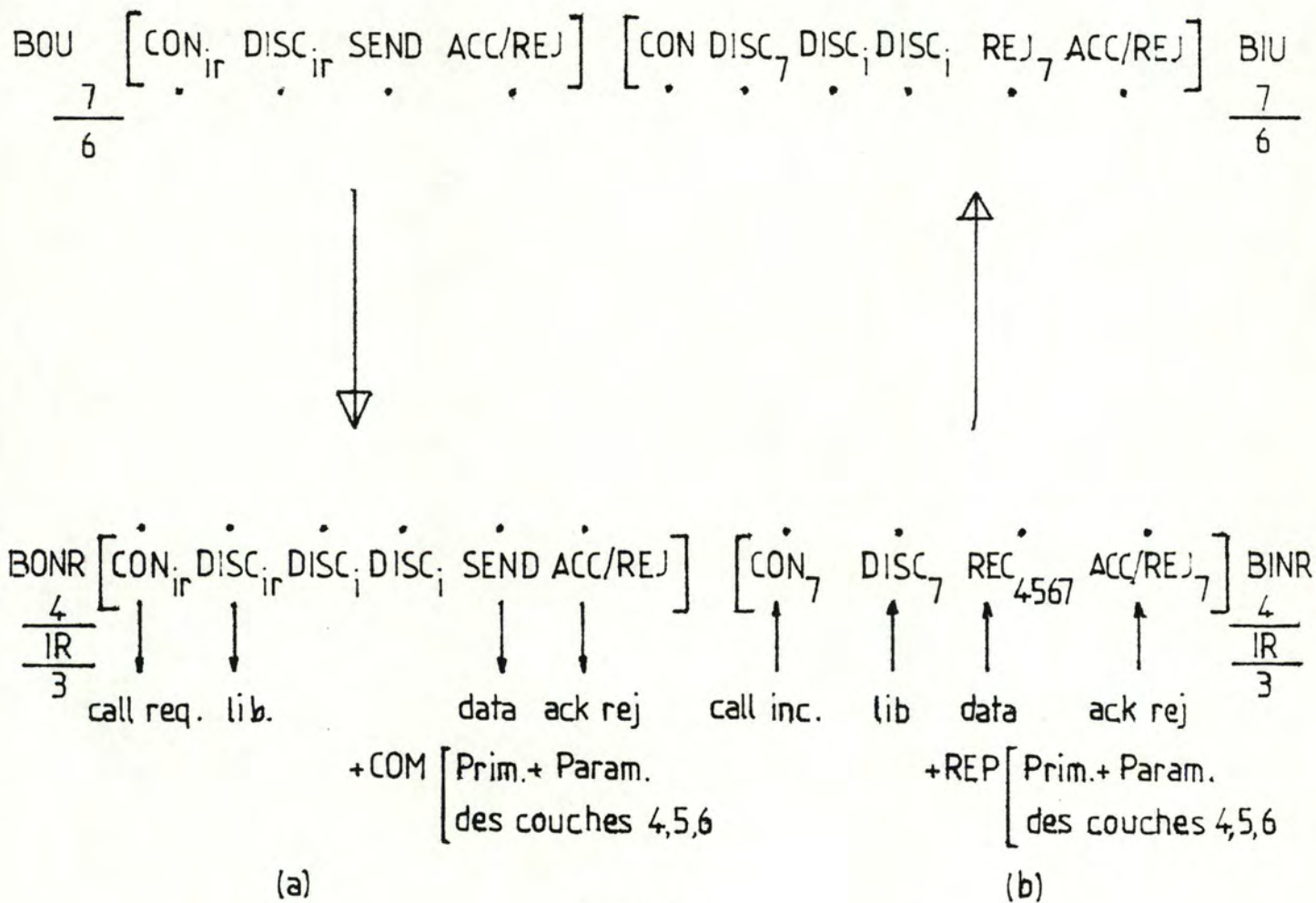


Fig 4.2.2.

Au paragraphe précédent, nous avons indiqué comment s'opère leur cheminement. Celui-ci dépendra de la fonction qu'il doit assurer et éventuellement des paramètres que l'octet d'information porte en complément.

- 2 - D'opérer la conversion des formats standards des informations (Appel, Libération, Données, Primitives de couches) de et vers les couches supérieures (Application) en des formats variables selon le type de réseau appelé à supporter ces informations et vice-versa.

Pour fixer les idées, les figures 4.2.3. a et b représentent la correspondance de formation des octets BONR et BINR à partir des primitives d'un réseau X25. Ces diverses conversions seront examinées au chapitre VI.

#### 4.2.2 - OCTETS D'INFORMATION EN ENTREE (BIU/IR) ET EN SORTIE (BOU/7)

-----

- EN ENTREE - (BIU de la figure 4.2.2 b)

-----

Le programme de l'utilisateur est susceptible de recevoir:

- la partie affichage, c'est-à-dire les différentes grilles ou menus que l'utilisateur aura à remplir à sa demande.
- les différents messages qui lui parviendront.



BINR  
 IR ← L3  
 STATUS BYTE ENTRANT

CON	.	call incoming	] de L3 vers L4
ACC	.	call acc.	
REJ	.	call rej.	
DIS	.	demande de liberation	
REC	.	data	
	.	ack (rr/rnr)	
	.	reinitialisation	
	.	reprise	
		Parametres de destination i=4,5,6 ou n=7	

(b)

BONR  
 L3 ← IR  
 STATUS BYTE SORTANT

CALL REQ	.	CON en traversée
LIB	.	DISC de n'importe quel niveau
DATA	.	SEND
ACC/REJ	.	
REINIT./REPR.	.	(uniquement en X 25)
		Parametres d'origine i=4 5 6 ou n=7

(a)

Fig 4.2.3.

Le niveau 7 est mis au courant d'une entrée et de sa nature par un byte BIU composé des bits suivants:

### 1. CON

ou Appel entrant

Le programme utilisateur est invité à aller chercher le contenu de ZL dans la zone de lecture (ou d'entrée) du moniteur, puis à le transmettre à l'application. (L'afficher à l'écran, y compris les facilités).

### 2. DISC

ou Libération

Voir CON.

(Afficher à l'écran le contenu de ZL, y compris les causes).

### 3. REC

ou Données structurées en

- SEGMENTS: pour X25: le moniteur se chargera de formater ZL.
- BLOCS de longueur à définir selon X21, asynchrone, Ethernet.

Dans tous les cas, ces longueurs ont fait l'objet d'une négociation au niveau 4. Ces données seront formatées à partir du contenu de ZL pour être transmises à l'utilisateur comme CON ou à certaines couches selon le résultat de l'analyse de la primitive de ZL par le moniteur.



#### 4. DISC\_i et DISC\_i+/-

ou indice de dérapage en cours de cheminement.

Ils seront transmis:

- à l'utilisateur pour information et
- au moniteur pour exécution.

Ils donneront lieu à une déconnexion après analyse de la cause par le moniteur dans les couches  $i$ ,  $i \pm 1$  incriminées.

#### 5. ACK ou REJ

Ces primitives positionnant un bit sont transmises à l'utilisateur (écran) pour lui indiquer que l'appel sortant qu'il a transmis est accepté ou refusé. Dans ce dernier cas, il est invité à s'intéresser au contenu de ZL pour en connaître les causes.

-----  
 - EN SORTIE - (BOU de la figure 4.2.2a)  
 -----

Lorsque l'utilisateur a enregistré les informations (via l'écran par ex.) dans l'ordre

- appel sortant
- envoi des données
- libération ainsi qu'accessoirement son autorisation ou son rejet d'un appel sortant,

comme pour l'entrée une double opération a lieu:

- 1 - Le positionnement des bits de l'octet BOU qui aura pour référence respectivement CON, SEND, DISC, ACC et REJ.
- 2 - Le transfert des informations sous forme:
  - d'un segment incomplet pour un appel sortant ou une libération.
  - de segments successifs pour l'envoi de données dans la zone ZE.

C'est le moniteur qui se chargera du contrôle.

Dans le cas d'envoi de données, le moniteur se chargera:

- 1 - de contrôler l'introduction par les couches 6, 5 et 4 de la mise d'un "en-tête" à chaque segment c'est-à-dire:
  - CDUI
  - CSUI
  - TDT (avec ses paramètres N (R)/N (S) fournis à l'émission et contrôlés à la réception par le module du contrôle des flux).



- 2 - d'orchestrer les opérations, et notamment que ZE soit complètement alimentée par le niveau 7, avant que les niveaux 6 (CDUI), 5 (CSUI), 4 (TDT) viennent placer en en-tête leur primitive de couche.

En fait, c'est après avoir transmis ses informations à ZE que le bit SEND de BOU, destiné aux couches inférieures, sera positionné. Toutefois, il y a lieu de remarquer que les informations qui vont de la source à l'utilisateur (écran par exemple) doivent être formatées par le moniteur pour être enregistrées dans ZE et que ce formatage est fonction du réseau de transmission.

#### 4.2.3 - OCTETS D'INFORMATION A L'ARRIVEE (BINR/7) ET AU DEPART (BONR/IR)

-----  
- A L'ARRIVEE - (BINR de la figure 4.2.2b)  
-----

Les bits de cet octet correspondent aux bits de BIU, mis à part l'éclatement de DISC lors de la traversée des couches inférieures à celles de l'application.

La signification des bits qui composent cet octet est:

- |        |   |
|--------|---|
| 1. CON | Indique à la couche réseau qu'un APPEL    |
| =====7 | ENTRANT ainsi que ses facilités a été     |
|        | enregistré, et cette information che-     |
|        | minera (Type A) en transit jusqu'à l'uti- |
|        | lisateur.                                 |

## 2. DISC

=====7

Indique à la couche réseau qu'une DEMANDE DE LIBERATION ainsi que ses causes a été enregistrée et cette information cheminera (Type A) en transit jusqu'à l'utilisateur.

## 3. ACC/REC

=====7

Indique à la couche réseau qu'une ACCEPTATION OU REJECT d'un appel sortant que la station a émise, a été enregistrée et cette information cheminera (Type A) en transit jusqu'à l'utilisateur.

## 4. REC

=====7

Indique à la couche réseau qu'une donnée sous forme de:

- PAQUETS X 25  
(ACC/REJ, CON, DISC, primit. + param. de L4, L5 et L6 dernier bloc de données partielles)
- MINI-SEGMENTS  
(primit. + param. de L4, L5 et L6 dernier bloc de données partielles)
- SEGMENTS  
(X 21 ou Ethernet)

a été reçue.  
Cette information cheminera jusqu'à l'utilisateur.



Les informations associées à ces bits de fonctions sont stockées en vrac dans une file d'attente du niveau 3 soit sous forme de:

- PAQUETS X 25
- FAUX SEGMENTS (X 21 ou Ethernet);  
ce ne sont ni des paquets, ni des segments parfaits c'est-à-dire de longueur de 128 bytes pour les premiers ou de la longueur de la TPDU négociée au niveau 4 (Ex.: 512 bytes).

Quoi qu'il en soit, un BINR de IR (bas), dès qu'on lui positionne un bit, fera une demande d'allocation de bus pour être transféré dans la couche IR (haut)(Fig.4.2.4)

Dès que ce transfert est exécuté, la couche IR (haut) sollicite (via une allocation du bus) de la couche IR(bas), le transfert

- du premier paquet (X25)
  - du premier segment(X21 ou Ethernet),
- de la pile de réception à la zone ZL.

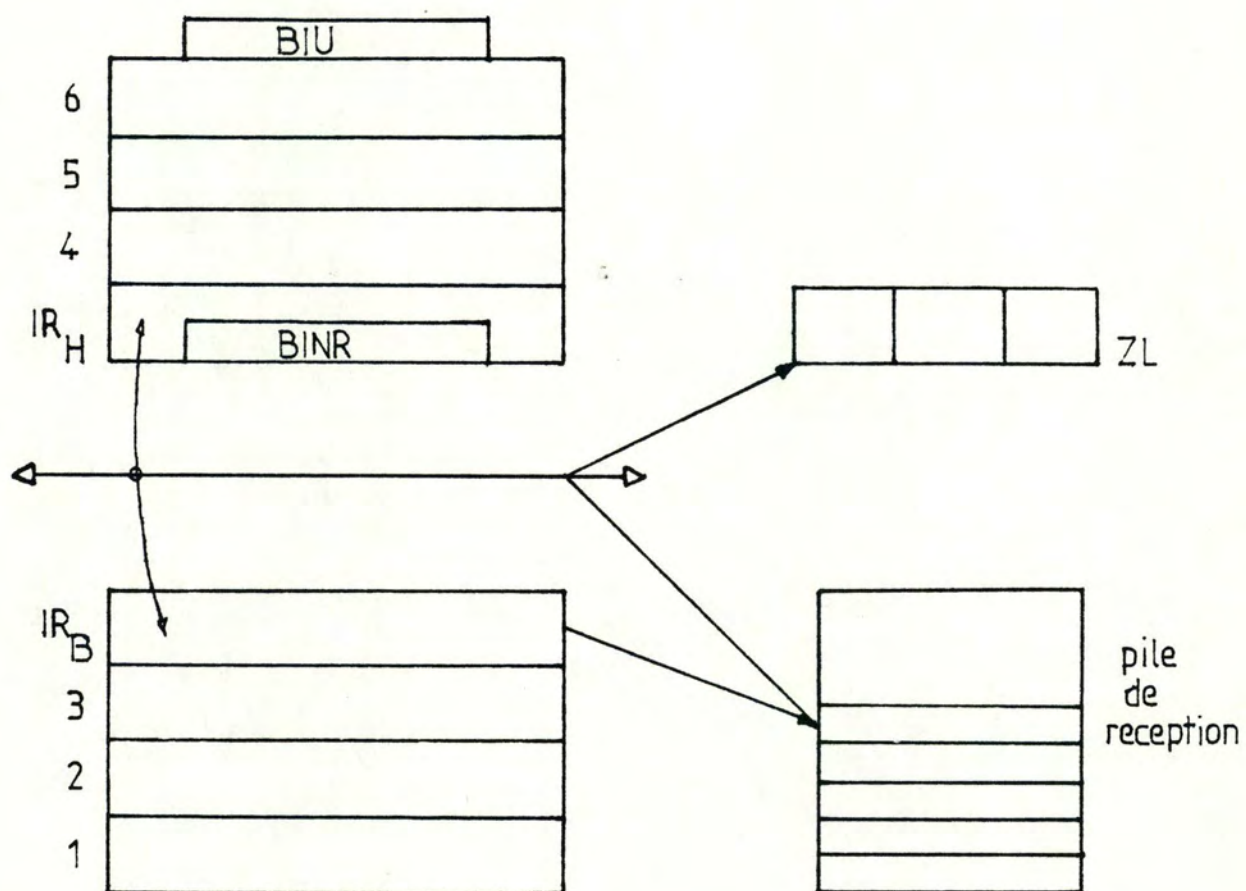


Fig. 4.24.

# NOTA BENE

-----

- 1 - Il serait utile en X21 et Ethernet, de normaliser la longueur des segments, quel que soit ce qu'ils contiennent(Ex.: 512 bytes).
- 2 - En X21, les informations échangées sur le réseau pourront être protégées par un protocole tel que le BSC.  
Au niveau 4, le contrôle du séquençement complètera ce contrôle de validité de bout en bout.
- 3 - A la mise d'un bit de BIU,l'utilisateur (parce qu'il s'agit d'un CON, DISC, ACC/REJ correspondant) est invité à prendre directement:
  - les paquets: si appel, libération... en X25
  - les mini-segments: si appel, lib... en X21, Ethernet.
- 4 - Si le bit de BIU correspond à un REC, le moniteur teste s'il s'agit:
  - a - D'une "primitive\_+\_paramètre"  
et en connaît le niveau n. Il transmettra alors le REC accompagné de la valeur n.  
Le test de validité de l'état présent du graphe constituera en fonction de l'état suivant, une réponse à la commande.
  - b - D'une "donnée pure"  
dont le bit significatif doit atteindre le niveau 7, mais après avoir transité successivement par les niveaux:



- 4: TDT pour le test de validité et de contrôle de  $N(R)/N(S)$ .
- 5: CSUI pour le test de validité
- 6: CDUI pour le test de validité.

Pour toutes ces opérations, il y a toujours le risque qu'une ou l'autre avorte, auquel cas il y aura renvoi vers la couche IR d'un DISC  $i$   
DISC  $i, i+1$   
qui sera transmis au moniteur.

-----  
 - AU DEPART - (BONR de la figure 4.2.2a)  
 -----

Les bits de cet octet correspondent aux bits de BOU mis à part l'éclatement éventuel du DISC de départ en 3 DISC:

1. DISC :

=====nr=====

Appel à la libération comme l'utilisateur l'a indiqué dans son BOU.

2. DISC :

=====i=====

A été généré lorsqu'en transit dans une des couches, un test de validité d'état s'est révélé négatif.

3. DISC

=====i+/-1:=====

A été généré lorsque la traversée de la couche i vers la couche adjacente (i - 1 ou i + 1) n'a pas été autorisée par suite de la présence d'un RNR.

En ce qui concerne les autres bits, c'est-à-dire ceux qui ont pour origine BOU et qui arrivent en BONR

- soit en transit direct (Type A)  
(ex. CON, DISC, ACC/REJ).
- soit en progression couche par couche  
(type B)  
(ex. SEND),

il y a lieu de considérer différents types d'opérations en considérant préalablement que le BOU a été transféré à la couche L3 ou L2 selon le réseau.

Ce transfert à cette couche y provoque une demande prioritaire de location du bus, pour indiquer à ce niveau du réseau qu'il y a une information à prendre (cfr plus loin pour le détail du mécanisme de transfert).

REMARQUE:

Il y a priorité à l'émission par le réseau. A la réception on admet une pile dans les limites fixées par la largeur de fenêtre gérée par les N (R)/N (S).

Cette information dans ZL est:

- soit un mini-segment  
(pour appel, libération, ack/nak, primitives + paramètres de L4, L5 et L6).
- soit un segment (données): ici le transfert se fera après contrôle et indication du séquençement (paramètres du TDT), tandis que son découpage s'opérera via le moniteur.



#### 4.2.4 - PROGRAMME DES COUCHES

Au sujet des programmes internes des couches 4, 5 et 6, nous pouvons spécifier, dès à présent, un certain nombre de caractéristiques à leur propos.

- a - Ils sont chacun contrôlés par un automate d'états, afin de gérer le séquençement de leurs actions.
- b - Ils disposent de leurs primitives propres.
- c - Ils entretiennent des dialogues de négociation et de décision avec les couches similaires des stations distantes.
- d - Ils contrôlent le cheminement des octets d'information entre les différentes couches 4, 5 et 6, ainsi qu'avec les deux extrêmes IR et 7 (Application).
- e - Ils disposent chacun d'un certain nombre de primitives de fonctions auxquelles, selon le cas, on adjoint des paramètres:
  - l - qui sont d'origines différentes et qui sont classés en catégories:
    - G: générés par le programme  
(ex. code de la primitive)
    - U: fournis par l'utilisateur  
(ex. facilités à l'appel)
    - S: fournis par le système  
(ex. date et heure)
    - E - pour ceux qui caractérisent l'environnement, c'est-à-dire l'équipement:
      - soit de traitement de données lié au niveau 7 des applications  
(terminal, ordinateur, processus...)

- soit de transmission de données  
lié aux niveaux 1, 2 , 3 des  
réseaux (X25, X21...). Ces paramè-  
tres sont assignés une fois pour  
toutes avant l'exécution des opé-  
rations.

Ex. Indicatif et classe du terminal,  
vitesse de transmission liée à la  
largeur de la fenêtre.

- D - pour ceux qui sont fournis par d'autres  
stations

Ex. Indicatif de l'appelé.

- M - pour ceux fournis par la station elle-  
même, soit enregistrés ou traités par  
le Moniteur.

Ex. Adresse de destination en valeur  
symbolique.

- 2 - qui sont de destinations différentes ré-  
pertoriées en:

- F - pour les informations destinées au  
module de contrôle de flux (Syst.)

- I - pour le contrôle du correspondant  
ou pour les informations des-  
tinées à une utilisation ultérieure,  
comme les caractéristiques du desti-  
nataire ou résultant de la négocia-  
tion.

Ex. Taille mémoire.

La topologie des programmes est illustrée aux  
figures 4.2.6a et b, où apparaissent les programmes  
principaux. Les autres modules (D, F, I...) sont  
inclus dans le moniteur.



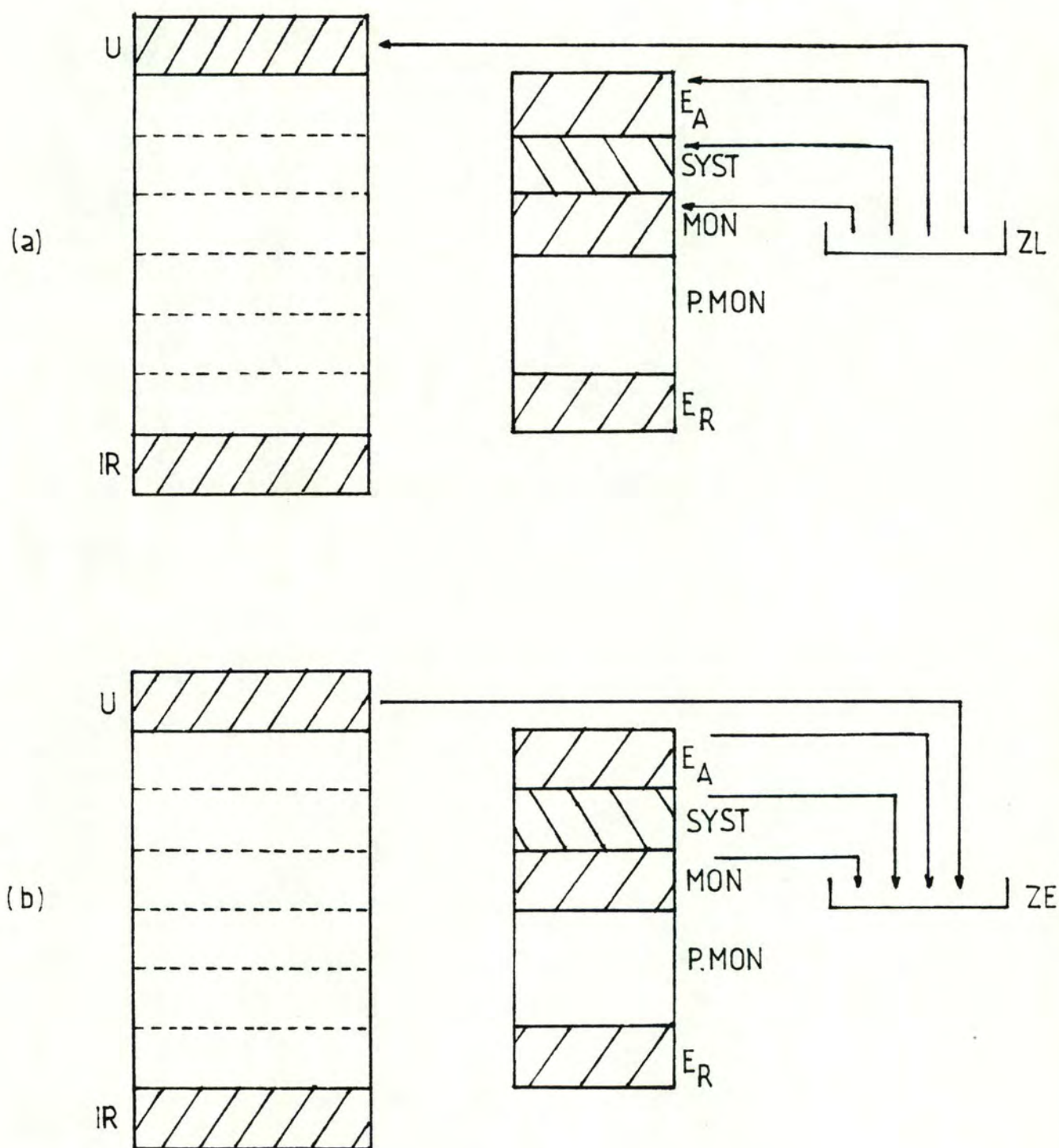


Fig. 4.2.6.



#### 4.2.5 - MECANISMES DES OPERATIONS

Ce paragraphe aura pour but de déterminer, de façon primaire, les fonctions des routines du moniteur de gestion et d'interruption.

Au niveau physique, les interruptions d'interface en entrée et en sortie assurent les relations

Ligne  $\begin{array}{l} \rightarrow \text{ZL} \\ \leftarrow \text{ZE} \end{array}$  ; puis les octets d'états

d'interface L3/L4 communiquent d'un niveau à l'autre la présence ou non d'une opération réalisée ou à réaliser.

Les paramètres ou informations sont transférés ensuite par le moniteur vers/au

- niveau 7
- réseau
- différentes couches: 4, 5 et 6.

-----

- EN ENTREE -

-----

La zone de lecture ZL (adresse fixe et connue) reçoit du réseau des paquets dont la validité et le séquençement ont été vérifiés par P (R)/P (S).

Une interruption I (provenant du DMA lié à HDLC) informe alors le moniteur que quelque chose a été reçu (paquets ou segments).

On pourrait envisager également la scrutation périodique du BIU, dont le positionnement d'un bit révélerait la même chose.

Au préalable, on constitue à partir du niveau 3, une file d'attente qui sert à la réception de segments.

La routine d'analyse du moniteur peut alors:

- 1 - Appeler le niveau 4 par l'octet d'information, afin de contrôler le séquençement par le T (R)/T (S) du segment. Il s'agit également d'effectuer un contrôle de flux par le module de ce niveau.
- 2 - Analyser principalement la validité du message reçu, qui ne peut être que:
  - une information venant du niveau 3 (CALL INC, INF, LIB, ...) qui constitue
    - .soit une réponse (à CALL REQ)
    - .soit un message nouveau (LIB).
  - une réponse aux commandes de L4, L5 ou L6, sous la forme respectivement de TCA, TCC, TBR..., RS<sub>X</sub>; RD<sub>Y</sub>... + paramètres.
  - des données qui concernent le niveau L7 ou éventuellement leurs ACK avec N (R) à contrôler.
- 3 - En ce qui concerne L3: positionner un bit du STATUS BYTE d'interface selon le message reçu.  
Une interruption du moniteur informera le niveau 4.
- 4 - En ce qui concerne les niveaux supérieurs à L3:
  - Pour les réponses, le test du code de la primitive va permettre l'envoi d'une interruption du moniteur vers le niveau concerné (4, 5, 6).



- Puisqu'il s'agit d'une réponse à une demande d'une primitive, cette interruption n'est autre qu'un RETURN dans le niveau à l'adresse de la fin du module de la primitive appelante. Ceci permet de poursuivre dans le graphe, en fonction du contenu de la réponse.
  - Etant donné le nombre divers des appels, des tables par niveau seront constituées, afin d'enregistrer les commandes attendant les réponses, ainsi que les adresses du processus.
- 5 - En ce qui concerne les données pures, le processus est similaire, mais l'envoi d'une interruption est destiné au programme d'application. Ce dernier recevra pour lui, la donnée dépouillée de ce qui a déjà été analysé.

En fait, c'est un pointeur qui sera directement transmis au niveau 7, et qui contient l'adresse de ZL.

#### REMARQUE:

Un accusé de réception au niveau des segments (pour les paquets, RR/RNR sont internes à X25) n'est envisageable que si la procédure réseau ne dispose pas d'un protocole de recouvrement d'erreurs (X21 ou Ethernet, par exemple). Par contre, la numérotation T (R)/T (S) pour assurer la vérification du séquençement et du flux au niveau 4 de transport sera prévue dans tous les cas.



Leur analyse va permettre, de plus, d'effacer l'ancienne donnée et d'enchaîner avec la suivante. La zone sortante où se trouve ZE en sera informée par message. En fait, il y a lieu d'opérer, dans tous les cas, les vérifications de ZL (validité, séquençement N (R)/N (S)...) et de les empiler pour les traitements précités, avec mise à jour des pointeurs d'adresse.

-----  
 - EN SORTIE -  
 -----

La zone d'écriture ZE (adresse fixe et connue) reçoit des informations des différents niveaux du programme, ainsi que les numéros de séquençement du niveau 4.

Lorsque ZE est rempli, le moniteur dont elle constitue une zone, en informe le niveau réseau L3 par l'intermédiaire de l'octet d'interface.

Prenons quelques exemples:

#### 1 - CALL REQUEST:

- a pour origine la primitive interne CON, elle-même générée par un OPEN-LINK (voir fig. 4.1.2)
- n'est possible que si il n'existe aucune connexion au niveau L3 (testé dans L4)
- transmis par L3, qui connaîtra les zones où se trouvent les informations complémentaires qui se situent au niveau:
  - \* 7 du programme d'application pour  $AD_1$  (réseau),  $AD_2$  (processus)
  - \* du moniteur (débit de l'unité, adresse source...).

- 2 - Une primitive de commande appartenant à l'un des niveau L4, L5 et L6, avec ses paramètres qui se situeront:

- \* soit dans le module du programme de niveau concerné - (N)
- \* soit en provenance de l'utilisateur - (U)
- \* soit dans le système (moniteur notamment) - (S).

Il s'agit par exemple de CSS, CDS, ... avec leurs paramètres.

Les origines d'une commande, dans le(s) processus ou dans le module de niveau seront, en fait, enregistrées. En effet, leur point de départ était une instruction permettant une interruption vers le moniteur. Une routine de celui-ci a pour fonction de déplacer des informations à partir de leur source (N, U, S) vers ZE. Un retour sera effectué chaque fois vers le point d'appel.

- 3 - Les données qui comptent plus d'un paramètre: Il s'agit de TDT (4), CSUI (5) et CDUI (6). Pour eux, l'opération précédente (interruption avec retour pour chaque niveau) doit, en plus, être complétée (par le moniteur) par le passage d'un niveau à un autre.

- 4 - Opération relative à ACC/REJ lors d'un appel.

Remarque: Il ne faut pas les confondre avec ACK/NACK décrits plus hauts qui eux, jouent le rôle de caractères de contrôle à l'intérieur du protocole de réseau. Dans tous les cas, il s'agira d'ajouter aux segments les S(R)/S (S). Ceci sera réalisé par le module contrôlant le séquençement, et responsable du protocole de bout en bout. Ce dernier prendra les dispositions qui s'imposent au niveau 4 (segment) pour opérer un RESET, un restart ou un DISCONNECT, selon la gravité du désaccord.



Quel que soit le cas, lorsque ZE est rempli, le réseau en sera averti par une interruption. Une routine réalisera alors les opérations de transfert du contenu de ZE vers le réseau. Celles-ci seront, bien entendu, précédées par le transfert de l'octet d'interface de IR vers L3.



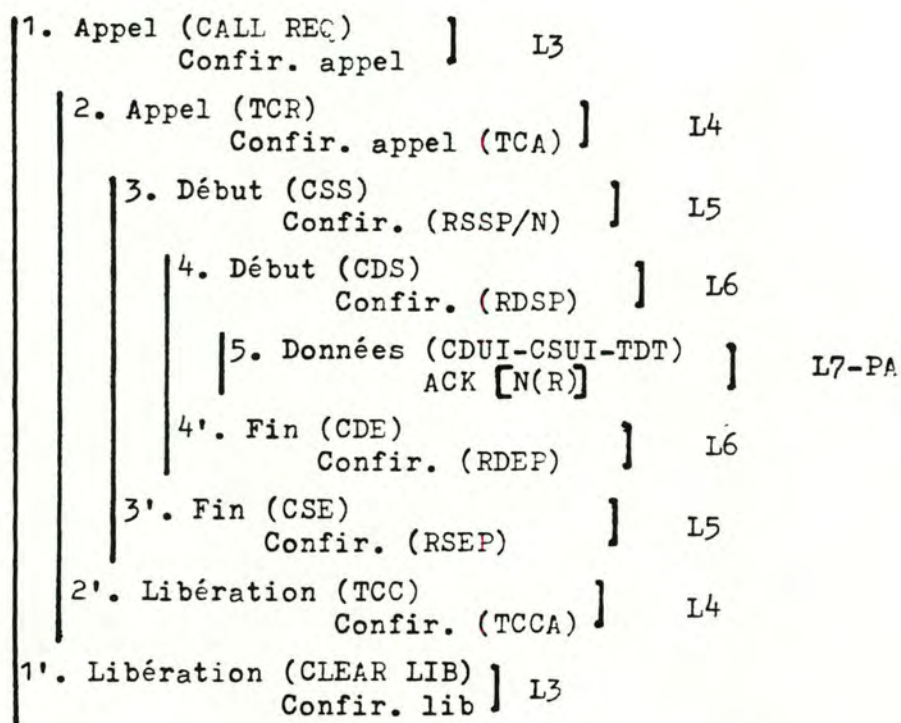
## 4.3 - IMBRICATION DES OPÉRATIONS

### 4.3.1 - SEQUENCE DE BASE DES OPERATIONS

Les échanges se font normalement dans l'ordre montant ou descendant des niveaux de protocole.

Il arrive parfois qu'il se produise des ruptures de séquence comme par exemple: un changement de sens de transmission, une reprise à un point de repère.... Quand cela se produit, la même règle s'applique pour la nouvelle séquence.

La séquence de base est détaillée à la figure 4.3.1.

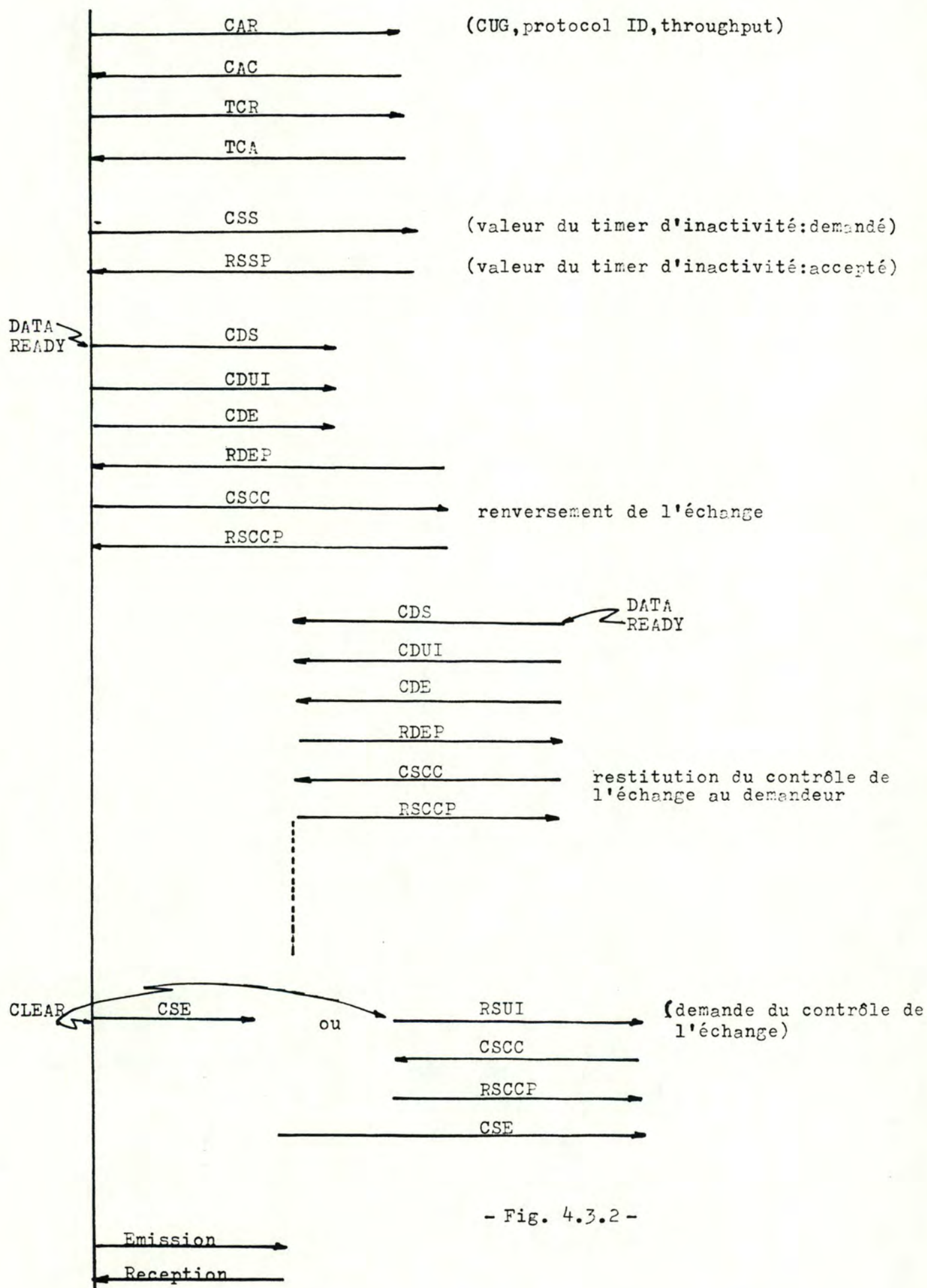


- Fig. 4.3.1 Séquence de base -

L'illustration d'un dialogue réel est donnée à la figure 4.3.2.

Il y a lieu de noter:

- a - que les primitives sont suivies de leurs paramètres.
- b - que chaque primitive plus ses paramètres ou données constitue un ou plusieurs segments, transformés en paquet puis en trame pour être transmis au réseau.



- Fig. 4.3.2 -



#### 4.3.2 - ILLUSTRATIONS DE SEQUENCES

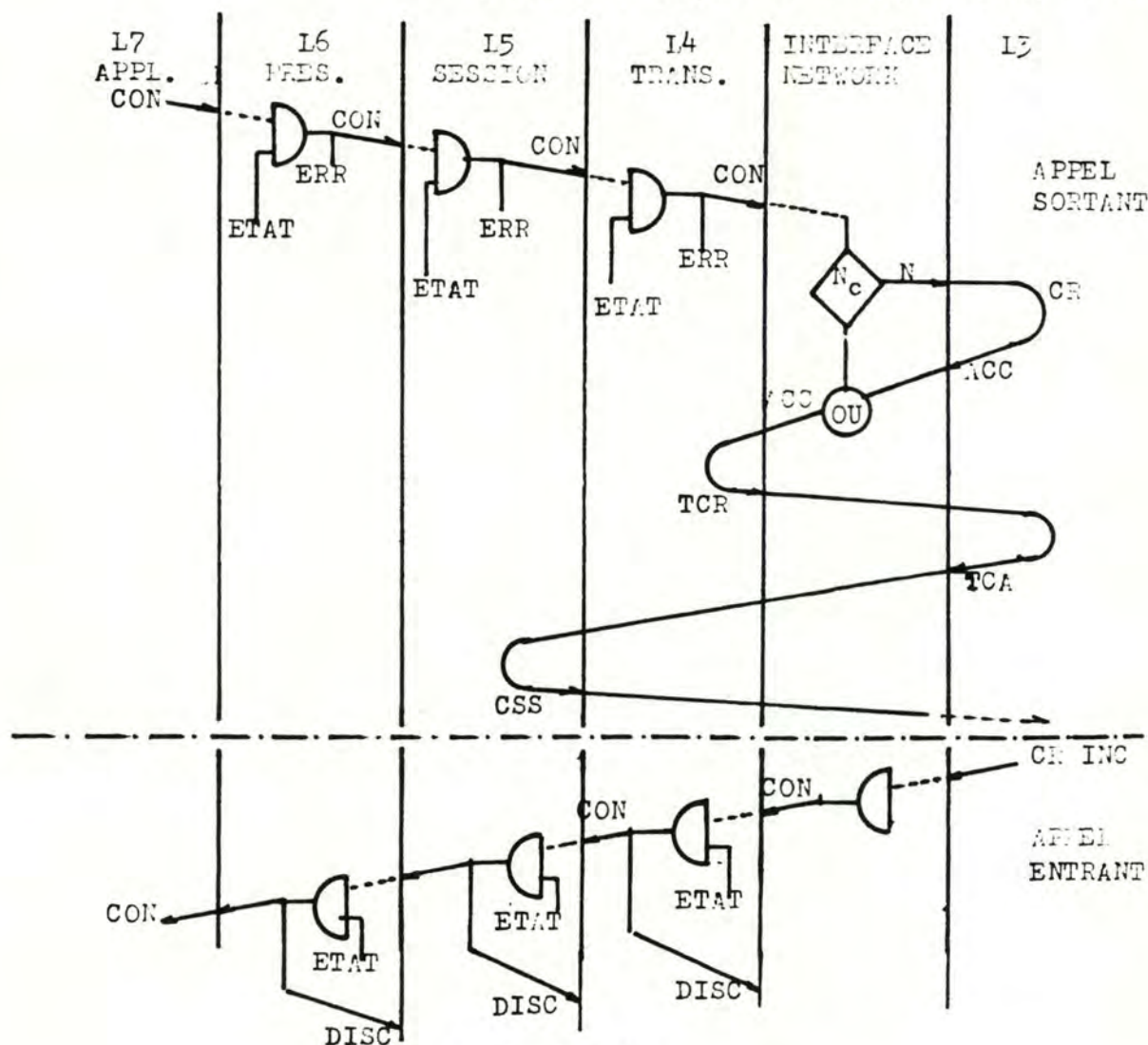
Ces illustrations, fort incomplètes, donnent une première idée de la chronologie d'ensemble des opérations pour les différentes couches: le § 2.2. n'explicitant au moyen de graphes, que les opérations relatives au niveau transport (4).

On trouve dans la suite, des échanges de messages plus particuliers, qui concernent respectivement les séquences Appel-Libération, le transfert des données en émission et en réception.

Quelques commentaires supplémentaires seront acquis dans la suite.

Rappelons:

1 - Pour une séquence d'appel (fig. 4.3.3)

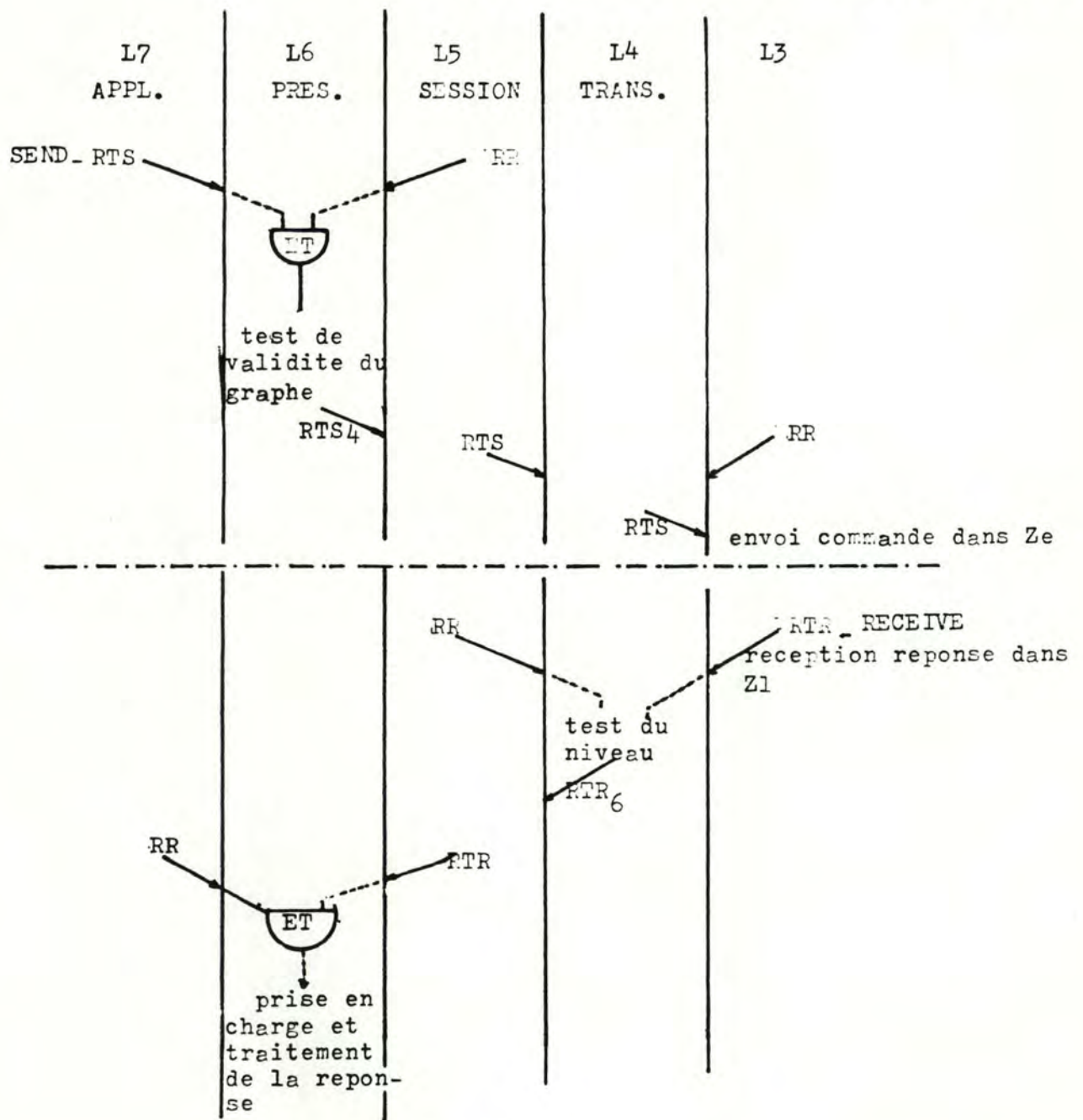


- Fig.4.3.3. -

La demande vient toujours du niveau le plus haut. La primitive CON descend alors les niveaux de proche en proche, pour autant qu'un test effectué dans chaque couche nous assure que cette dernière est à l'état de "repos".

Dans le cas où l'on recommence une nouvelle session sur la même liaison, il n'est pas obligatoire qu'un appel soit à nouveau nécessaire au niveau 4.

2 - Pour une séquence de données (fig. 4.3.4).



- Fig. 4.3.4 -



Le processus fondamental est que, de proche en proche, un SEND ou un RECEIVE se propage vers le bas ou vers le haut, de la même manière que précédemment. La conjonction de deux primitives (RTS. RR; RTR. RR) présentes entre les couches, permet le test de la phase ad-hoc du graphe, et l'écriture (ZE) ou la lecture (ZL) de la primitive plus les paramètres du niveau concerné.

- 3 - Il y a lieu de noter que la traversée complète des couches dans un sens ou dans l'autre n'est indispensable que pour des raisons de contrôle. Nous verrons plus loin qu'il est souvent nécessaire d'entrer ou de sortir des niveaux 5 et 6 et de ne transiter que dans 1, 2 ou 3 couches.

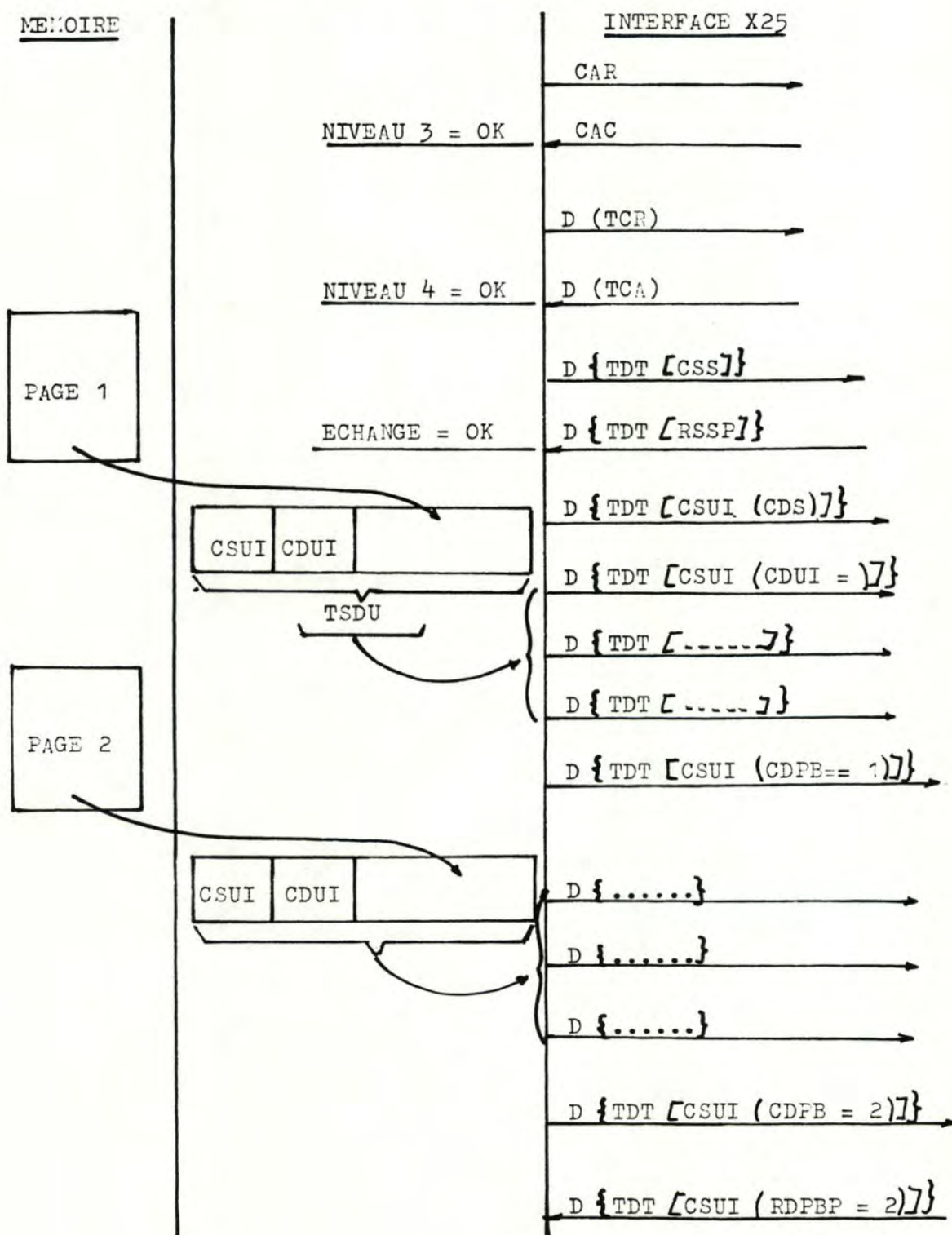
Une autre illustration d'une séquence fondamentale de transfert de données est représentée à la figure 4.3.5.

Pour la compréhension, nous noterons deux choses :

- 1 - Seul l'échange de données permet le cumul des primitives et des paramètres des différents niveaux (T: pour le transport (4), CS: pour la session (5), CD: pour la présentation (6)). Les primitives d'accusés de réception qui font écho aux primitives de commandes n'interviennent que couche après couche pour: l'appel, la libération, le renversement du sens de communication...



MECANISME GENERAL D'ECHANGE D'UN TEXTE.



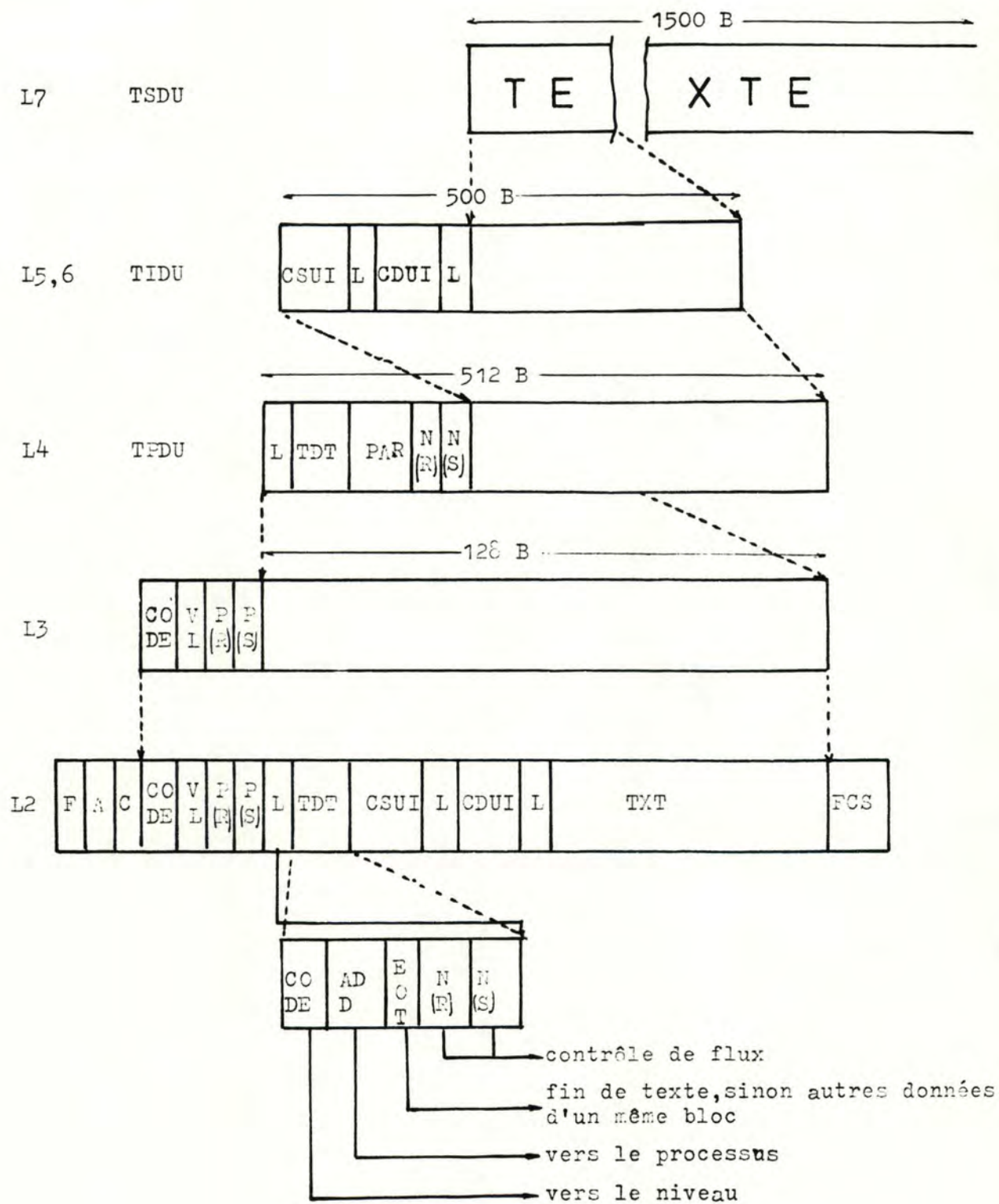
- Fig. 4.3.5 -

- 2 - Les primitives paramétrisées tant de commandes que de réponses relatives aux données s'imbriquent, niveau par niveau, les unes dans les autres comme l'indique la figure 4.3.6.

Parallèlement à ces insertions, il y a découpage successif du bloc physique en segments (niveau 4) puis en paquets (niveau 3).

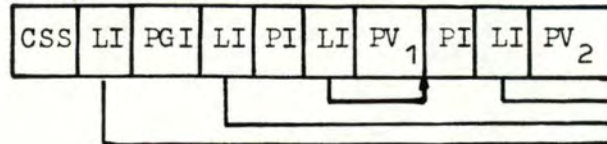
Précisons, pour terminer, qu'en ce qui concerne les paramètres des primitives, ils se structurent (fig. 4.3.7) en groupes (PGI) et chacun d'eux en paramètres séparés (PI). Chacun des éléments étant obtenu pour l'analyse, par un découpage basé sur l'enregistrement systématique de sa longueur (LI).

Nous reverrons tout ceci en détail dans les chapitres suivants.

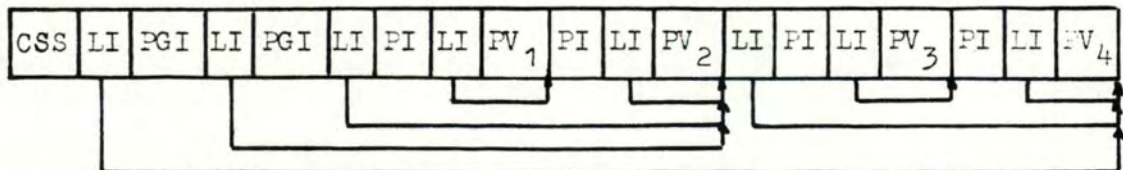


- Fig. 4.3.6 -





1 PGI, 2 PI  
 PV<sub>1</sub> = identité appelé  
 PV<sub>2</sub> = date, heure



PV<sub>1</sub> = identité appelé  
 PV<sub>2</sub> = date, heure  
 PV<sub>3</sub> = W appellant  
 PV<sub>4</sub> = mode TMA, TES...

- Fig. 4.3.7 -

## CHAPITRE V - FONCTIONNEMENT PARTICULIER DES COUCHES

=====

### INTRODUCTION

A - Généralités

B - Fonctions des différentes couches supérieures

### 5.1 - NIVEAU TRANSPORT

5.1.1 - Fonctions générales

5.1.2 - Fonctions détaillées

5.1.3 - Contrôle de flux

5.1.3.1 - Principes

5.1.3.2 - Contrôle de flux

5.1.4 - Contrôle de bout en bout

5.1.5 - Fonctions spécifiques du protocole

### 5.2 - NIVEAU SESSION

5.2.1 - Fonction générale

5.2.2 - Fonctions détaillées du protocole

### 5.3 - NIVEAU PRÉSENTATION

5.3.1 - Importance de cette couche

5.3.2 - Fonctions générales

5.3.3 - Fonctions détaillées des protocoles possibles

5.3.3.1 - Classe 0

5.3.3.2 - Classe I

5.3.3.3 - Classe II

5.3.3.4 - Classe III

5.3.3.5 - Classe IV

### 5.4 - APPLICATION

5.4.1 - Primitives du niveau 7

5.4.2 - Formats

## CHAPITRE V

---

### INTRODUCTION

#### A - GENERALITES

Nous considérons comme acquis que le logiciel de chacune des stations d'un réseau local est structuré en 7 couches. Chacune d'elles communiquent avec ses voisines en faisant dire par les primitives d'intercouches: "je désire...", "je dispose...".

Il existe donc:

- 1 - La couche "réseau" divisée en 3 niveaux dont les fonctions sont:

##### Niveau\_1\_ou "Physique":

gère la transmission de/vers le réseau au moyen d'organes physiques (portes, registres,...)

##### Niveau\_2\_ou "Liaison":

assure la transmission après avoir établi une liaison physique (permanente ou virtuelle).

Ex. la trame en HDLC.

Cette couche est nécessaire pour assurer une liaison physique à faible taux d'erreurs.

##### Niveau\_3\_ou "Réseau":

commute dans le réseau et assure la parfaite chronologie des opérations à tout moment (appel, échange de données, libération) même en cas d'anomalies, si le protocole le supporte.



2 - Une couche intermédiaire (Interface Réseau ou Network Layer).

Celle-ci n'est présente et indispensable que si le réseau joue, dans la station, un rôle différent du rôle de simple support d'échanges des informations des couches supérieures. Ce sera le cas lorsque plusieurs types de réseaux doivent être compatibles avec celles-ci ou lorsque des portes (gateways) sont ouvertes vers d'autres réseaux.

3 - Des couches réparties sur 4 niveaux dont les fonctions seront largement détaillées plus loin, mais pour lesquels on peut déjà dire:

Niveau\_4\_ou "Transport":

joue un rôle charnière entre les couches inférieures (1, 2, 3) et supérieures (4, 5, 6). Etant donné son rôle, il dispose de nombreuses fonctions, dont plusieurs ont déjà été citées.

Les plus fondamentales seront explicitées dans ce chapitre.

Niveau\_5\_ou "Session":

gère les échanges entre stations (terminaux, utilisateurs), sous des aspects divers qui diffèrent selon les partenaires, la nature des échanges ou le contenu fonctionnel des informations.

Niveau\_6\_ou "Présentation":

qui, après négociation des possibilités de l'émetteur et du récepteur, assure l'adaptation et la mise en page des messages d'information destinés à l'utilisateur.

Niveau\_7\_ou "Application"

C'est le niveau de l'utilisateur qui disposera des données à l'émission et à la réception.

L'utilisateur peut être un opérateur, un programme, un processus, une machine.

B - FONCTIONS DES DIFFERENTES COUCHES SUPERIEURES

Chaque couche possède ses primitives (commandes et réponses) avec chacune ses paramètres. Ceux-ci appartiennent à deux catégories fonctionnelles:

- 1 - Ceux qui négocient entre les processus appelant et appelé, les valeurs qui seront retenues pour la couche supérieure adjacente.

Le principe est le suivant: conjonction ou alignement sur la valeur la plus faible ou la moins riche.

Ex. Mise en page, taille mémoire...

- 2 - Ceux qui permettront à la couche:

- a - d'initialiser les échanges

Ex.: indications du service concerné (Vidéotex, télétex...)

- b - d'identifier les relations

Ex.: correspondants appelant/appelé, numéro du document...

- c - de remplir ses fonctions

Ex.: le couple N (R)/N (S) contrôlera le séquençement et le flux.

L'objectif poursuivi dans ce chapitre 5 est de procéder à l'analyse complète des spécifications normalisées qui permettront:

- de comprendre dans le détail le rôle et les fonctions des différentes couches supérieures.
- de passer à la phase de réalisation et de programmation.

Nous examinerons de la sorte et dans l'ordre, les chapîtres suivants:

- transport (4)
- session (5)
- présentation (6)
- application (7).



## 5.1 - NIVEAU TRANSPORT

### 5.1.1 - FONCTIONS GENERALES

Bien qu'elles possèdent beaucoup de points communs, certaines fonctions définies par l'ISO (fig. 5.1.1.1), par l'ECMA (fig. 5.1.1.2) et par le CCITT (fig. 5.1.1.3) sont absentes chez l'un ou chez l'autre.

De plus, l'ECMA les classe selon la catégorie de l'équipement de transmission de données.

# COUCHE TRANSPORT

	Fonctions ISO	DNA	SNA	DSA	CNA	P ou p	P document
1	Appel et libération d'une liaison	SCL/NSL	PC	TL	TS	P	TCR/TCA TCC/TCCA
2	Adressage	TL	PC/TC	TL	TS	p	
3	identification	SCL		TL		p	
4	transfert normal de données					P	TDT
5	Contrôle de flux	NSL	PC	TL	TS	p(u)	
6	Protocole de bout en bout	NSL	PC ITC	TL	TS	P	
7	Segmentation	NSL	PC	TL	TS	p	
8	Concatenation					p	
9	Multiplexage	NSL	PC	TL	TS	p	
10	T S D U	NSL	PC	TL	TS	p	
11	Expédition de message	NSL				P	
12	Détection et recouvrement d'erreurs	NSL	PC	TL	TS	P	TBR
13	Purge					P	
14	Qualité de service Notification		PC	TL	TS	p	
15	Indépendance du réseau						
16	Classe de service		PC	TL	TS		

SCL: Session control layer

TC : Path control layer

TS : Transnet service

NSL: Network service layer

TC : Transmission control

P/p: Primitives/paramètres

P/document: primitives pour transfert de documents.



## E C M A

## CLASSES DES TERMINAUX ET SERVICES

CLASSE 0	TERMINAUX ÉLÉMENTAIRES
CLASSE 1	TERMINAUX FONDAMENTAUX
CLASSE 2	TERMINAUX À CONTRÔLE DE FLUX
CLASSE 3	TERMINAUX À RECOUVREMENT D'ERREURS
CLASSE 4	TERMINAUX À DÉTECTION ET RECOUVREMENT D'ERREURS (DATAGRAMME)

N°	SERVICES	0	1	2	3	Commentaires
1	Segmentation	X	X	X	X	Découpage en segments pour transmission - rapidité et contrôle (BCC/CRC petits tronçons) TSDU → TIDU → TPDU Bloc physique      segments      paquets terminés par END MARK
2	Détection d'erreurs = -corruption -duplication -séquencement -délivrance fautive	X				Erreur - de procédure - invalidité TPDU (code...) - invalidité paramètres - $N(R) \neq N(R)$ attendu d'où selon la gravité: - TPDU ERREUR - clôture - classe 0 - RJ commande - N(R) - classe 3 - Recouvrement - classe 3 - Purge - Reprise à N(R) classe 2,3
3	Libération permet la déconnexion par le niveau 4		X	X	X	- T C C uniquement pour l'appel en classe 0 - T C C (Dis. Request) + TCCA (Dis. conf.) permet l'appel et la libération en classes 1, 2, 3



N°	Services	0	1	2	3	Commentaires
4	Multiplexage		x	x	x	Alternance des segments de blocs vers des destinations différentes sur une même ligne physique
5	Concaténation		x	x	x	Plusieurs blocs dans un segment = $\sum$ TSDU
6. 7.	Contrôle de flux + AK permet de freiner le flux des niveaux > ou l'arrêt complet en diminuant CDT $\rightarrow$ 0			x CDT fixe	x CDT variable	Contrôle de flux mais aussi de l'erreur
8	EXPEDITED DATA + AK Messages courts envoyés en priorité			x	x	Envoyé même si CDT = 0 Réception d'un AK de contrôle
9	PURGE PR + PC Congestion et perturbation du réseau avec pertes de données			x	x	<ul style="list-style-type: none"> <li>• <math>N(R) = N(S) = 0</math>, TSDU = 0</li> <li>• Réappel du TSDU des disquettes</li> <li>• Nouveau CDT faible</li> <li>• Numérotation reprend à zéro au départ de la fenêtre</li> </ul>
10	Recouvrement d'erreurs cfr. 3				x	Selon la gravité - RESTART avec/sans nouvel appel - RJ commande (12) - TPDU Erreur - termine + cause
11	Erreur TPDU	x	x	x	x	Message de clôture fautive avec raisons
12	RJ commande $N(R) \leq N(R)$ attendu				x	Commande de RJ avec $N(R) = N(R)$ acquitté + 1 d'où on effectuera la reprise

- Fig. 5.1.1.2 (suite) -

C. C. I. T. T.

- A - Etablissement d'une connection de transport
  - identification de la connexion
  - multiplexage de la connexion de transport
- B - Transfert de données
  - commande de séquence
  - détection d'erreurs
  - correction des erreurs
  - segmentation et combinaison
  - commande de flux
  - purge

Remarques:

- 1 - Les fonctions ci-dessus ne sont pas toutes disponibles dans le service de transport de base.
- 2 - Pendant l'établissement d'une communication de transport, il y a négociation sur la base des règles suivantes:
  - le terminal appelant indique la classe de service de transport et/ou les fonctions facultatives requises
  - les fonctions sont différentes selon la classe de service
  - le terminal appelé indique la classe de service de transport et/ou les fonctions facultatives qu'il est prêt à accepter
  - tous les paramètres à utiliser dans la connexion de transport doivent être indiqués explicitement, faute de quoi des valeurs par défaut seront utilisées.
- 3 - Les fonctions du service de transport sont groupées (pour la négociation) dans un système hiérarchique de classes; les classes les plus élevées de la hiérarchie mettent en oeuvre toutes les fonctions des classes inférieures ainsi que les fonctions supplémentaires identifiées pour chaque classe.



### 5.1.2 - FONCTIONS DETAILLEES

Les principales fonctions proposées par les normalisations internationales sont explicitées ci-après:

#### a - Contrôle de flux

Ce contrôle fera l'objet d'un paragraphe spécial.

#### b - Contrôle du séquençement de bout en bout

Ce contrôle fera l'objet d'un paragraphe spécial.

#### c - Segmentation

Les blocs physiques sont fragmentés en segments.

Suivant que L3:

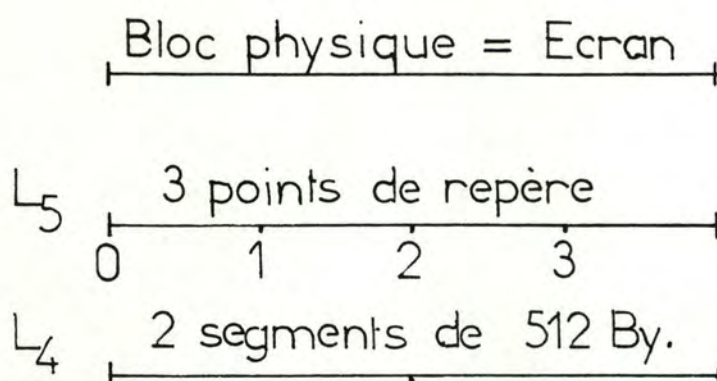
- est RCP: ils se découpent (ou s'associent) en paquets.
- n'est pas RCP: ils sont transmis (ou reçus) tels quels au réseau.

Notons que, comme les paquets, les segments disposent du jeu  $N(R)/N(S)$ , associés au cadre d'une fenêtre négociée au niveau 4, afin de contrôler le flux et le séquençement. Ceci fait également partie du protocole de bout en bout.

Remarquons enfin que:

- + si le réseau n'est pas RCP, alors il y a:
  - Détection d'erreurs
  - Reprise
    - au niveau des segments L4: Il n'y a pas de ACK. Dès lors les segments sont numérotés modulo comme les paquets.
    - au niveau du texte L5, grâce aux points de repère fixés à l'émission.





+ si le réseau est RCP, alors il y a :

- Détection d'erreurs

- Reprise

- au niveau L3 -par RNR

- par réinitialisation ou reprise par L3 ou L4

- au niveau L4 -par primitive TBR de L4

- pour des erreurs constatées à la réception. TBR demande le renvoi n fois à l'émetteur.

- par TPDU ERROR pour des erreurs de primitives ou paramètres pollués à la réception.

#### d - Concaténation

Plusieurs TPDU pourront être concaténés dans un TSDU du réseau. Le découpage se fera à l'arrivée, selon la longueur du premier octet et l'adresse de la destination au niveau transport.

Un TPDU de données pourra être concaténé avec un caractère de contrôle (par exemple du protocole BSC).

### e - Multiplexage

Le multiplexage, c'est-à-dire le transfert de message de un ou plusieurs programmes d'application de/vers des destinations différentes, ne pose pas de problème particulier, étant donné :

- 1 - qu'il existe dans tous les paquets une adresse de destination et des tables de rendez-vous à l'émetteur et au récepteur.
- 2 - que l'émetteur doit travailler en synchronisme. C'est le contrôle de flux, tant interne qu'externe (si RCP), qui conduit dans les cas extrêmes au freinage de certains programmes d'application.
- 3 - que la demande de crédit règle également le débit vers le récepteur. Elle a donc une influence à l'émission.
- 4 - qu'un moniteur de gestion et d'interruption assure le débit optimal aussi bien en entrée qu'en sortie.

### f - TSDU transfer

Le TSDU doit pouvoir indiquer comment un bloc physique peut être reconstitué, malgré le découpage en segments, paquets ou fragments. Si le TSDU contient un END MARK égal à 1, alors les TPDU n'ont pas atteints la longueur du bloc. Il faudra que le END MARK du TSDU soit égal à 0 pour indiquer que la longueur du bloc est atteinte.

### g - Envoi d'un télégramme

Le service télégramme permet à un processus d'être à tout moment à l'écoute des autres.

Ici, un TSDU sur une connexion de transport, ne sera pas soumis à faire la queue, à des restrictions de ressource ou au contrôle normal de données. Il est transmis très rapidement.



Il devrait donc pour se faire comporter:

- 1 - deux primitives d'intercouches spéciales:  
SEND-TELEG et RECEIVE-TELEG.
- 2 - des primitives entre stations:
  - TG - TXT : envoi
  - TG - ACK : reçu
  - TG - ERR : non reçu.

Ces primitives ainsi que leurs paramètres sont spécifiques à ce service.

#### h - Récupération d'erreurs

Selon la gravité de l'erreur ou la classe des services, la récupération d'erreurs comportera:

- l'envoi de TPDU ERR ou TBR pour un TPDU invalide (erreur de procédure, de séquençement ou encore la non validité des paramètres),
- l'envoi à partir du point de repère fixé par L6.
- l'envoi d'un REJECT avec N(R) indiquant à partir de quel segment il y a lieu de les renvoyer.
- une réinitialisation (RESET) avec rétablissement de la connexion au niveau 4 et effacement accompagné de reprise au niveau du segment.
- une reprise (RESTART) avec rétablissement des connexions aux niveaux 3 puis 4, effacement et reprise au niveau du bloc(si possible).

#### i - Purge

Entre stations, un "PURGE REQUEST" peut être émis pour fautes graves. Un "PURGE INDICATION" accuse réception et prépare l'opération.

Il correspond à une réinitialisation, mais au niveau de la connexion transport. En d'autres termes, il y aura lieu de réitérer l'appel au



niveau transport (TCR/TCA), et de reprendre au début du segment précédent, avec destruction des données précédentes. Cette opération est donc analogue à un RESET, avec un rétablissement au niveau transport.

Il y a lieu de ne pas confondre avec le RESTART ou REPRISE, pour laquelle il y a CLEAR - Rétablissement de la connexion RESEAU, puis de la connexion TRANSPORT. Toutefois, cette opération ne peut préjuger de conséquences, favorables ou non, de sorte qu'en général, la session ne peut se poursuivre et qu'il est nécessaire de la suspendre par un "ABORT". Le principe est, en effet, que selon l'origine de l'erreur et de la procédure de récupération, il faut distinguer:

- niveaux supérieurs: récupérable
- niveau transport
- niveaux inférieurs: en général irrécupérable.

L'opération de PURGE est parfois confiée exclusivement au niveau L3.

#### j - Classes de service

Les classes de service sont données au paragraphe précédent pour le cas d'utilisation des terminaux. Elles s'étendent à d'autres équipements de données selon les performances qu'on lui demande de réaliser.

### 5.1.3 - CONTROLE DE FLUX

#### 5.1.3.1 - Principes

A - Examinons comment se transmettent dans une ligne, les données entre un émetteur et un récepteur.

Le débit s'écoule sans difficultés lorsque la vitesse de la ligne exprimée en Bytes/sec

- est supérieure au nombre de Bytes/sec que l'émetteur transmet
- est inférieure au nombre de Bytes/sec que le récepteur peut absorber, avec l'utilisation d'un ou de deux tampons d'interface.

Si ces deux conditions ne sont pas respectées simultanément, la ligne freine le débit trop élevé de l'émetteur, mais elle ne peut forcer la cadence d'absorption trop faible du récepteur (fig. 5.1.3.1.2), avec comme conséquence une file d'attente à chaque bout.

Si l'une ou l'autre condition n'est pas respectée, la pile se situe soit à l'émetteur (fig. 5.1.3.1.3) soit au récepteur (fig. 5.1.3.1.4).

B - Dans le cas de la figure 5.1.3.1.3 où le plus généralement on ne peut freiner le flux entrant par des signaux électriques, on le fera par message (sinon la pile grossira dans la mémoire principale (MP) car le goulot se situe en aval).

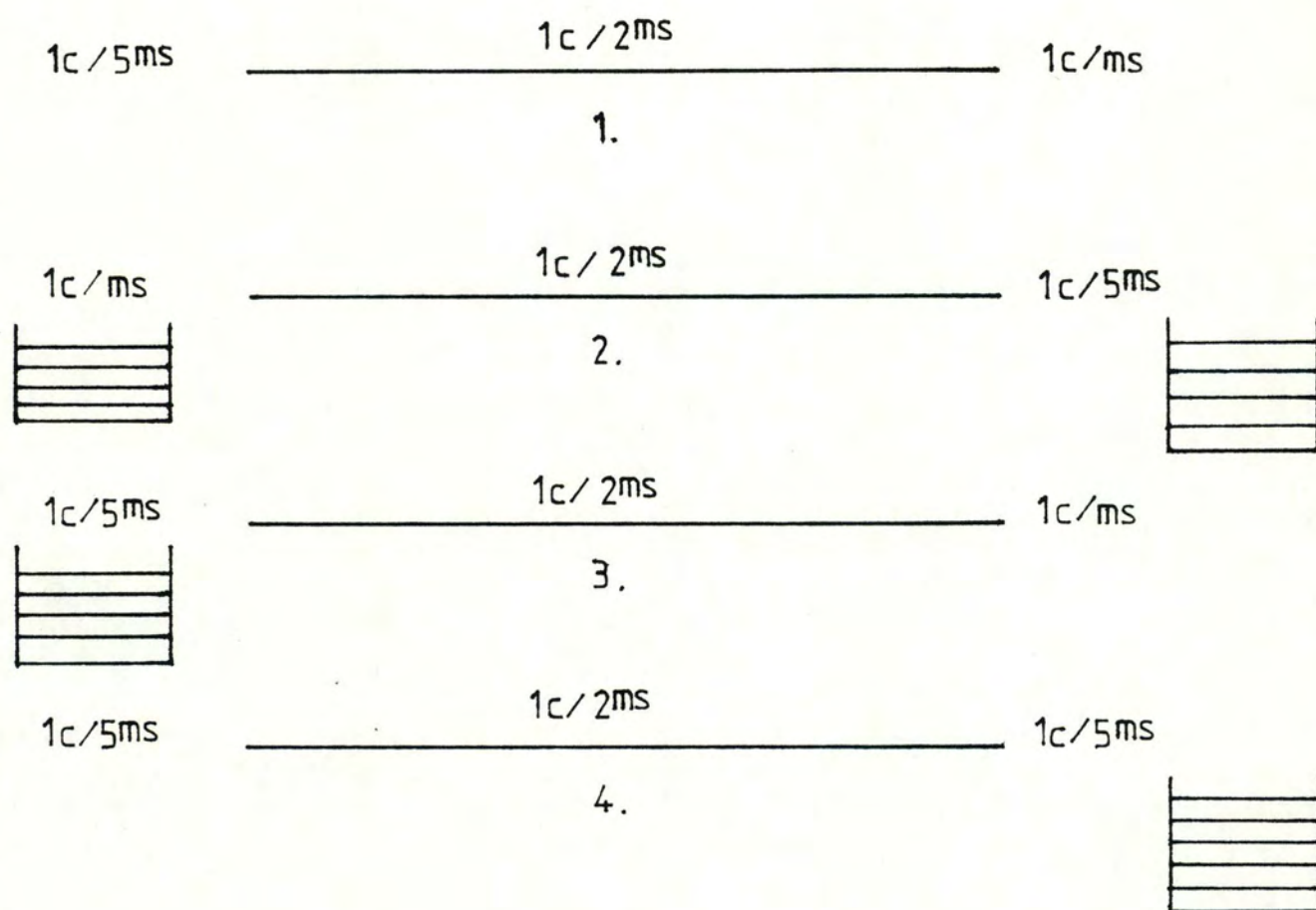


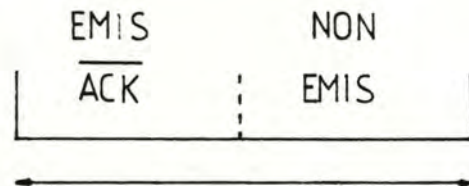
Fig. 5.1.3.1.



Ce message sera:

- a - s'il y a un protocole élémentaire: un crédit de  $n$  blocs transmis par le récepteur à l'émetteur.
- b - s'il existe un protocole de haut niveau (Ex: HDLC): un renvoi à l'émetteur d'un accusé de réception ACK sous la forme d'un RNR ( $n$ ) suivi d'un RR( $n$ ) si le seuil de la pile le permet.

La valeur de  $n$  est limitée par l'ouverture d'une fenêtre limitant le nombre de messages émis sans accusé de réception.



$W$  = largeur de fenêtre

D'une manière générale, ces contrôles sont basés sur le choix au départ

- 1 - d'un type de liaison au niveau L4 après l'envoi établi au niveau L3.
- 2 - de l'octroi du CREDIT. Celui-ci peut être:
  - a)- individuel
  - b)- statique
  - c)- dynamique.

Remarquons qu'on est souvent limité aux extrêmes a) ou c).

Le choix entre un des trois fait l'objet d'un paramètre de TCR.

- a - individuel: c'est-à-dire du type STOP AND WAIT, avec ACK au coup par coup.

- b- statique\_mais\_renouvelable. Ce crédit est réajusté périodiquement au niveau session en tant que paramètre d'une primitive de celle-ci. Généralement, sa valeur est égale au nombre de buffers circulaires disponibles. Il faut noter enfin que les ACK sont numérotés, mais non limités par une fenêtre.
- c- dynamique: La valeur de ce crédit est ajusté et initialisé comme ci-avant, mais ici, elle modifie la valeur de l'ouverture de la fenêtre. Il restreint ou augmente le nombre de segments que le récepteur peut recevoir en rafale sans accusés de réception. En général, ce crédit varie de 0 à 128.  
Il se peut, enfin, que le freinage puisse aller jusqu'à l'arrêt complet de la transmission (  $w = 0$  ).

- 3 - Les contrôles de bout en bout et de flux sont garantis par le jeu des  $N(R)/N(S)$ , attribués aux segments dans le cadre d'une fenêtre de largeur déterminée et négociable.

Rappelons que le contrôle de bout en bout permet de vérifier le séquençement des segments valides reçus, tandis que le contrôle de flux nous assure qu'à la réception, tout segment trouvera un tampon disponible.

- C - Dans le cas de la figure 5.1.3.1.2, la ligne freine le débit par ses REQUESTS, rythmés par la cadence de la ligne, tant du côté émetteur que du côté récepteur.



S'il y a poussée de la part du programme d'application, celle-ci sera asservie à la vitesse de la ligne. Si cet asservissement ne peut se faire, une pile sera prévue dans la mémoire principale de l'émetteur. C'est le cas de la figure 5.1.3.1.4.

Pour ce cas, la parade à la congestion par file d'attente à l'émetteur sera le freinage à la source, c'est-à-dire au programme d'application. Tout freinage en aval n'est efficace que si le débit important à l'émission est continu et non temporaire. Dans ces conditions, on pourrait réguler le débit en adoptant différents moyens:

- Par un contrôle physique au niveau du modem, en inhibant la borne "demande pour émettre" vers l'équipement de transmission de données.
- Par message, sous la forme d'un caractère  $X_{ON}$  ou  $X_{OFF}$  qui permet ou non l'émission.
- Le plus sophistiqué, est de réagir vers l'amont, à partir du niveau transport vers le programme d'application, en réduisant la fenêtre du niveau 4.

D - Le cas de la figure 5.1.3.1.2 est celui où la capacité insuffisante de la ligne conduit à la formation de files d'attente, tant à l'émission qu'à la réception. Les remèdes proposés plus haut seront donc appliqués.

#### 5.1.3.2 - Contrôle de flux

On peut comme on le fait au niveau 3, remplacer le crédit individuel en STOP and WAIT par des accusés de réception globaux au niveau des segments, soit:



- dans le cas d'une fenêtre de largeur négociable au niveau 4.
- $T(R)/T(S)$ , selon la règle équivalente à celle illustrée par les figures des deux pages suivantes.

# N U M B E R I N G      M E C H A N I S M

I-frame	0	0	N(s)	P	N(R)	
RR-frame:	1	0	0	0	P/F	N(R)
RNR-frame:	1	0	1	0	P/F	N(R)
REJ-frame:	1	0	0	1	P/F	N(R)

N (S)    =    send sequence number  
 N (R)    =    receive sequence number  
 V (S)    =    send state variable  
 V (R)    =    receive state variable

Modulo 8 numbering

(DCE = data circuit end    =    terminaison d'un circuit de données)

## 1 - DCE has 4 counters

- V (S) and received N (R) for I-frames sent by DCE
- V (R) en sent N (R) for I-frames received by DCE
- In a transmitted frame if  $N (S) = V (S)$  and  $N(R) = V(R)$   
     V (S) is incremented with 1 after transmitting an I-frame
- V (R) is incremented with 1 after receiving an Z-frame.

## 2 - Checking on the receipt of a frame

- Correct if  $N (S) = V (R)$  and if last received  
      $N (R) \leq N (R) < V (S)$
- If  $N (S) \neq V (R)$  the DCE sends a REJ with  $N (R) = V(R)$
- If N (R) incorrect DCE sends a FRNR

## 3 - Window Mechanism

More than k unacknowledged I-frames may not exist.  
 If  $V (S) = \text{last received} + k$  there is no further  
 transmission of Z-frames.

There is a DCE-action if:

1. V (S) received N (R) and time-out====> retransmission of the unacknowledged Z-frame initiated.
2.  $V (R) \neq \text{last transmitted } N (R)$   
     the DCE has still some Z-frames to acknowledge.



V(S)	Received N(R)	V(R)	Sent N(R)		V(S)	Received N(R)	V(R)	Sent N(R)
2	2	0	0	SABM	0	0	2	2
0	0	0	0	UA	0	0	0	0
1	0	0	0	N(S) I(0,0)	0	0	1	0
1	1	0	0	N(R) RR(1)	0	0	1	1
2	1	0	0	I(1,0)	0	0	2	1
3	1	0	0	I(2,0)	0	0	3	1
3	3	1	0	I(0,3)	1	0	3	3
4	3	1	1	I(3,1)	1	1	4	3
4	4	1	1	RR(4)	1	1	4	4



#### 5.1.4 - CONTROLE DE BOUT EN BOUT

Ce contrôle ne porte pas sur la validité des données au niveau 4, qui ont dû être contrôlées au niveau du réseau. Il porte sur le contrôle du séquençement des données, des primitives et des paramètres de celles-ci.

Le contrôle de ces erreurs se prolonge également par un protocole dont la technique de recouvrement est analogue à celle de l'HDLC, en utilisant, soit:

- la numérotation T (R)/T (S) qui a servi également au contrôle de flux.
- la fenêtre.

Il interviendra également de la même façon si X25 entame une procédure de réinitialisation ou de reprise qui se fera donc en synchronisme aux niveaux 3 et 4, l'une opérant sur des paquets et l'autre sur des segments.

### 5.1.5 - FONCTIONS SPECIFIQUES DU PROTOCOLE

Comme nous l'avons vu, les fonctions que le protocole aura à remplir sont nombreuses, et dépendront de la CLASSE du système de transmission de données. Toutefois, on peut les diviser en deux catégories:

- celles relatives à la négociation et qui seront exploitées par les couches supérieures.
- celles relatives aux fonctions de base et exécutées spécifiquement par la couche 4 de transport.

Nous citerons par exemple:

- Négociation: classe, mapping des adresses (si il existe plusieurs processus), segmentation...
- Fonction de base: contrôle de bout en bout, messages de contrôle, crédits...

Le protocole de transport se caractérise comme tous les autres par:

- l'usage de primitives dont le sens de chacune est précisé par des paramètres
- l'enchaînement de ces primitives dans des dialogues parfaitement spécifiés comme nous allons le voir.

1 - On trouvera aux figures suivantes tout ce qui concerne les primitives et les paramètres:

- Fig. 5.1.5.1 : la liste des primitives et des paramètres fondamentaux tels qu'ils ont été retenus par le CCITT.



- Fig. 5.1.5.2: la mise en oeuvre séquentielle pour l'appel, le transfert de données et la libération (à l'exception de TBR qui constate les anomalies).

2 - On trouvera aux figures suivantes tout ce qui concerne le séquençement des opérations, c'est-à-dire l'ensemble des graphes relatifs au niveau 4, et prenant en considération les primitives de cette couche et celles d'intercouche (spécifiées par l'octet d'information).

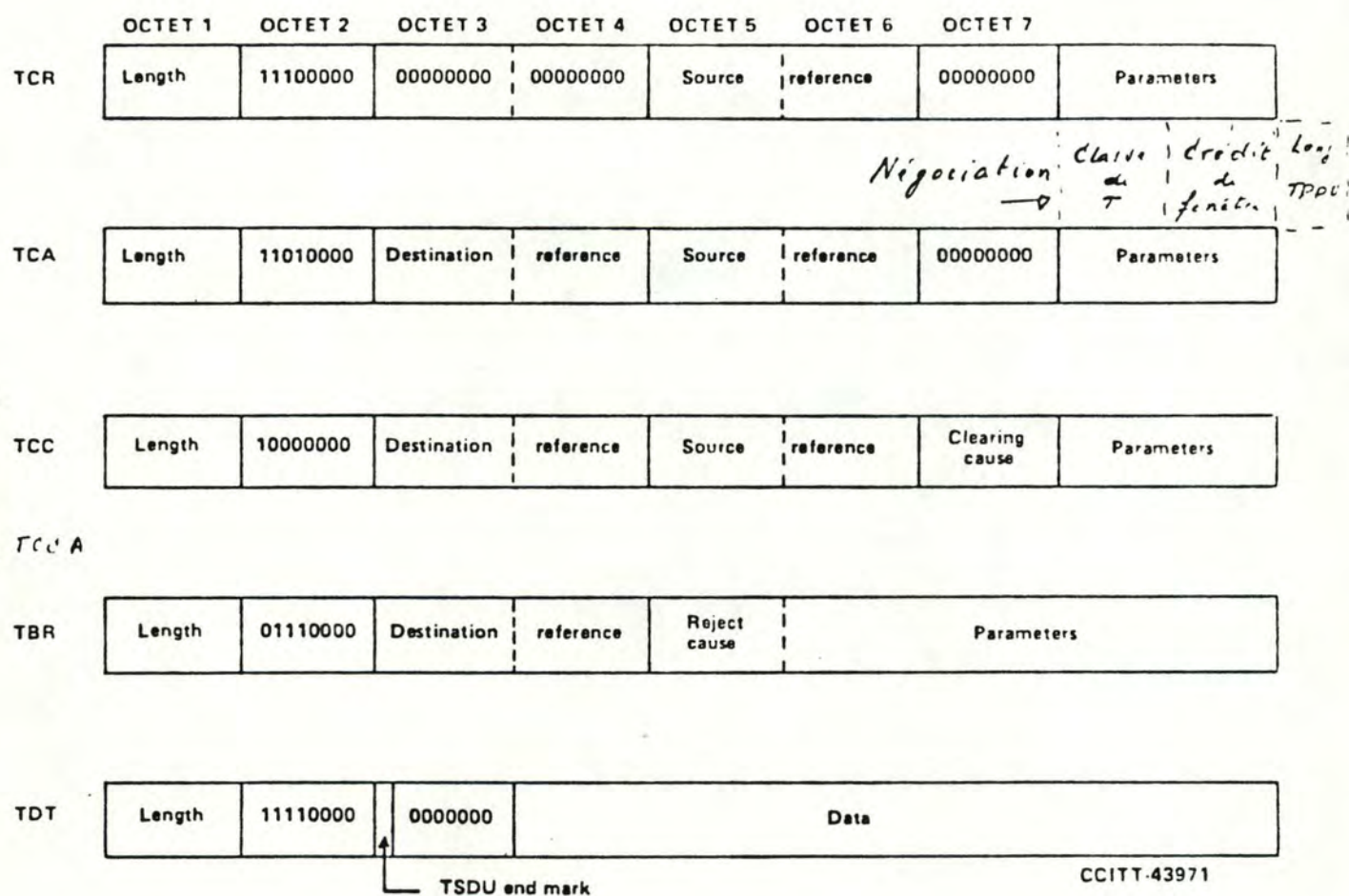
- fig. 5.1.5.3: graphe correspondant à la phase d'initialisation.
- fig. 5.1.5.4 : graphe correspondant à la phase d'établissement et de libération: côté appelant.
- fig. 5.1.5.5: graphe correspondant à la phase d'établissement et de libération: côté appelé.
- fig. 5.1.5.6: graphe correspondant à la phase de transfert de données: émission.
- fig. 5.1.5.7: graphe correspondant à la phase de transfert de données: réception.

3 - On trouvera en annexe B des figures complémentaires relatives au niveau 4 de transport:

- Les tables séquentielles équivalentes aux graphes. La présentation de ces tables est directement accessible à la réalisation et à la programmation.

- Les graphes relatifs aux stations appelante et appelée en ne retenant que les primitives propres.
- Les tables séquentielles traduisant les graphes précédents en automate qui dans le programme de la couche 4 de transport, contrôlera le séquençement des échanges.





BLOCS UTILISES POUR LE PROTOCOLE DU NIVEAU TRANSPORT

Fig. 5.1.5.1

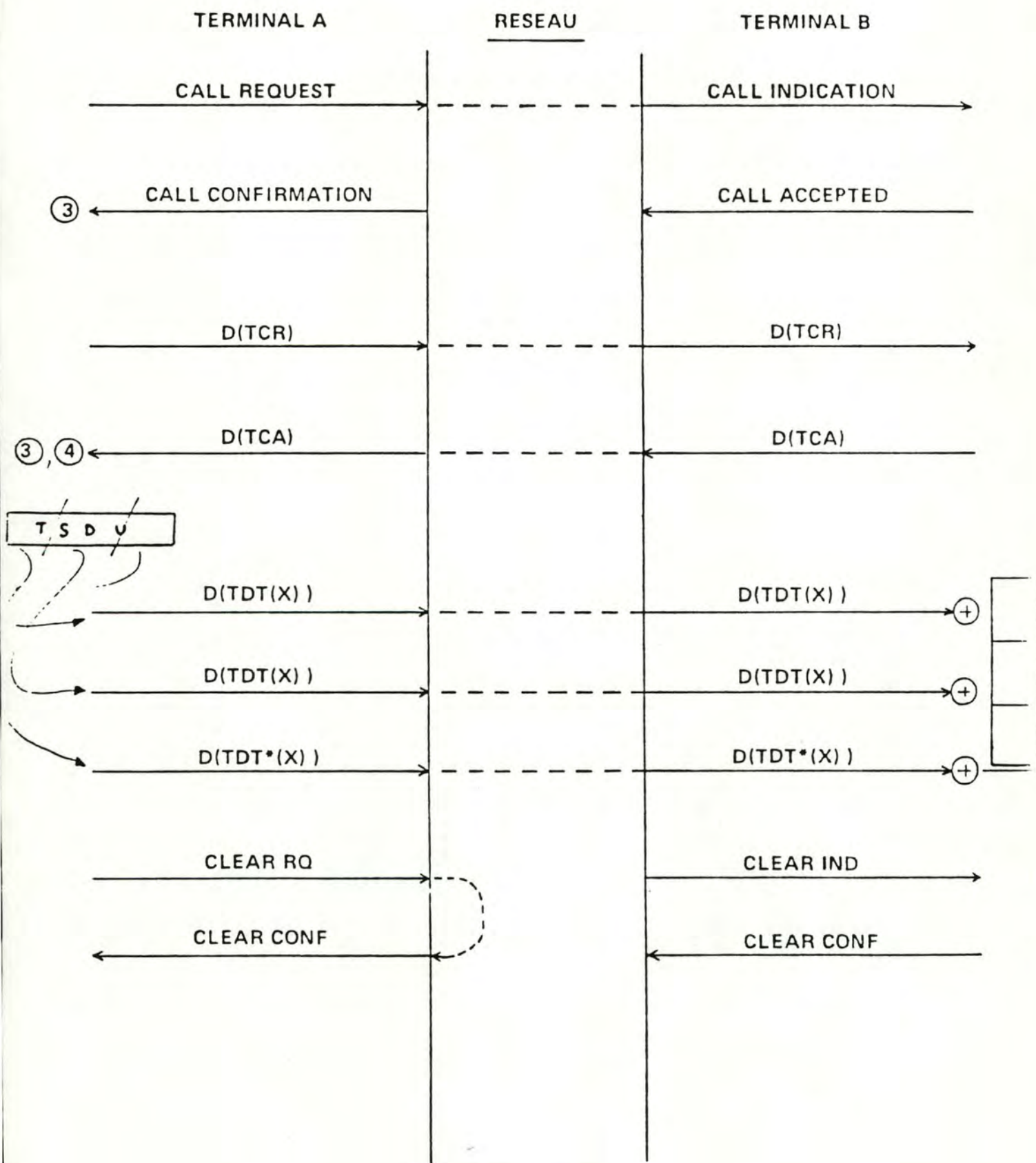


Fig. 5.1.52



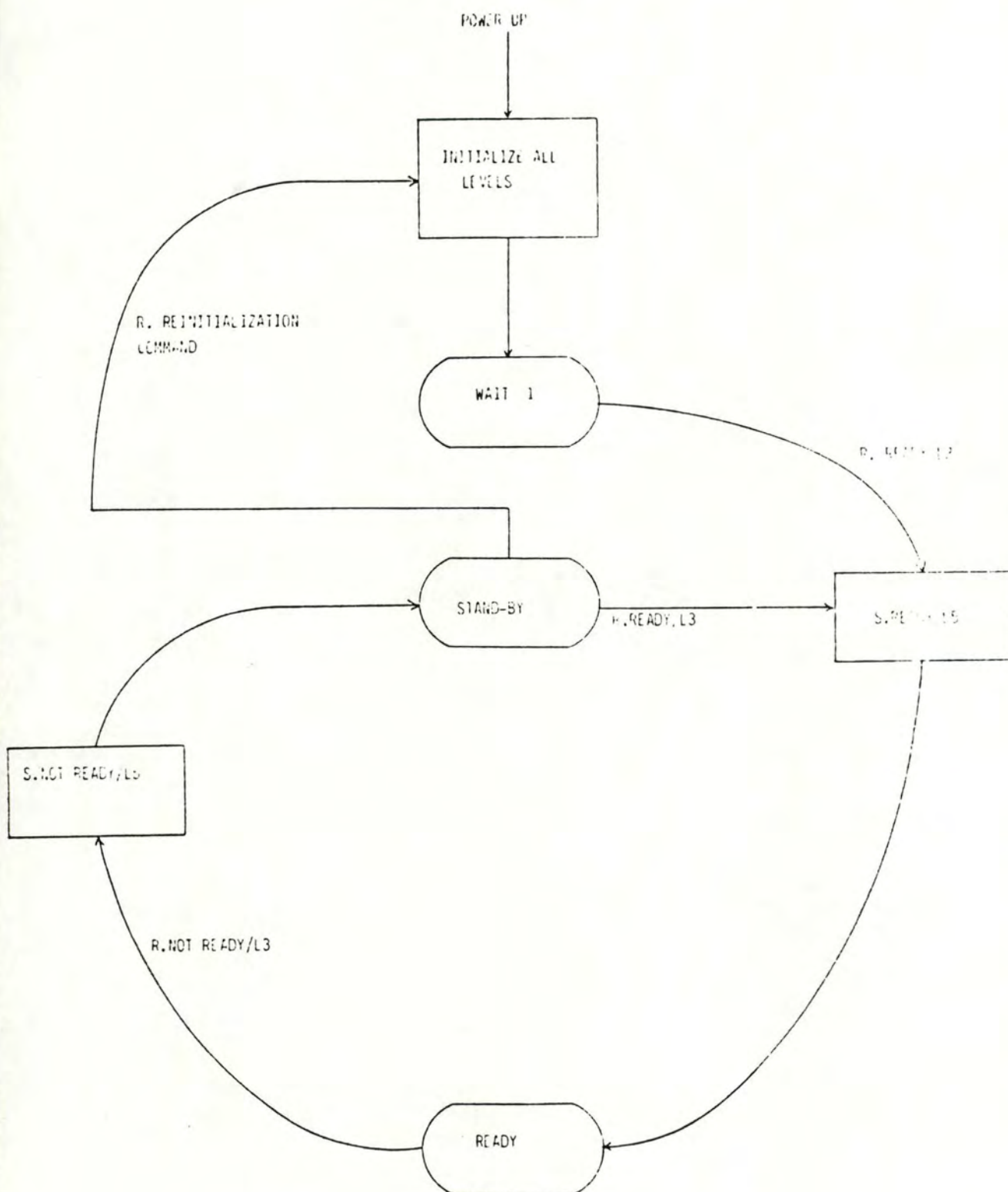


Fig. 5.1.5.3 INITIALIZATION  
GLOBAL FOR THE TERMINAL



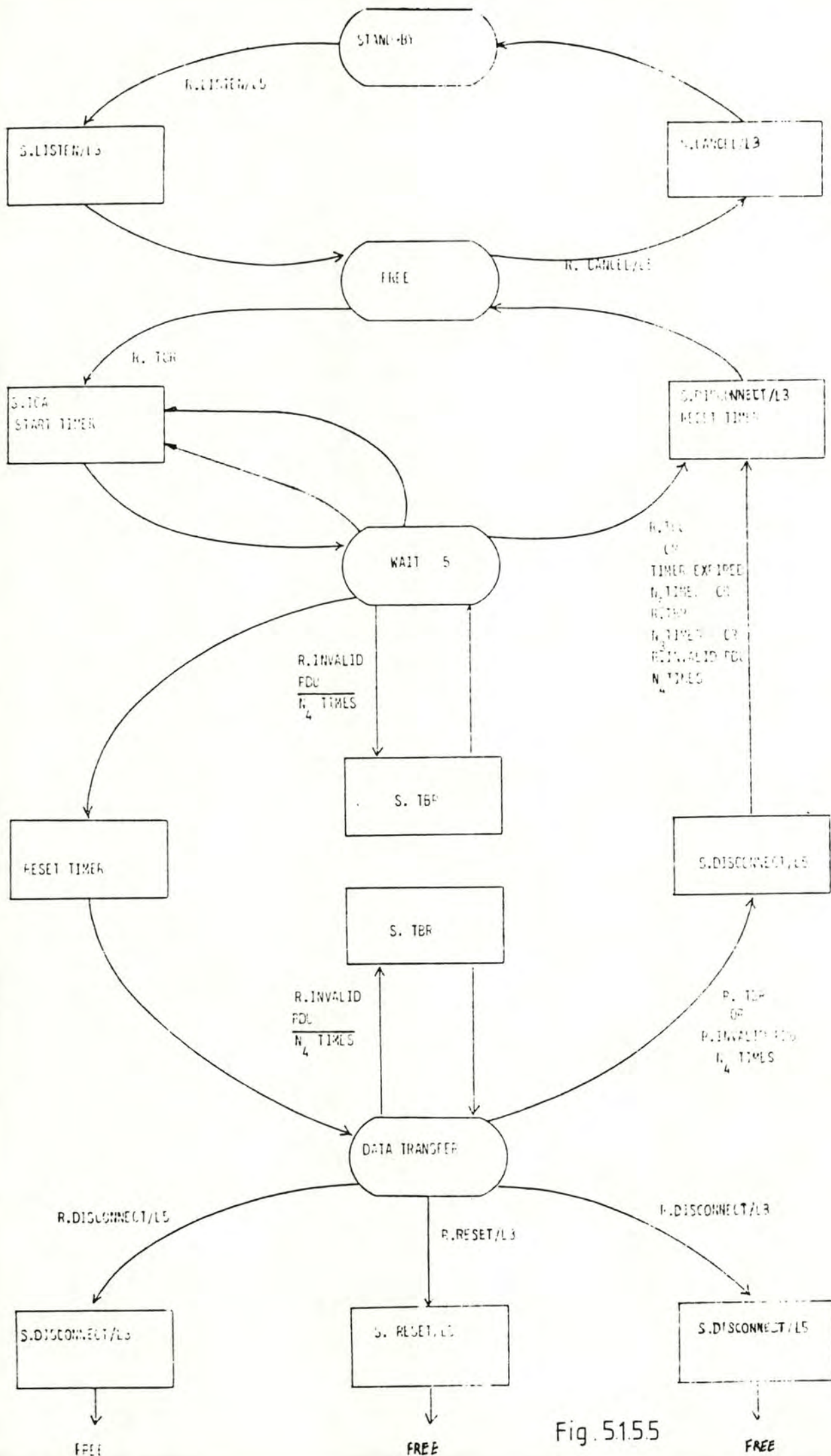
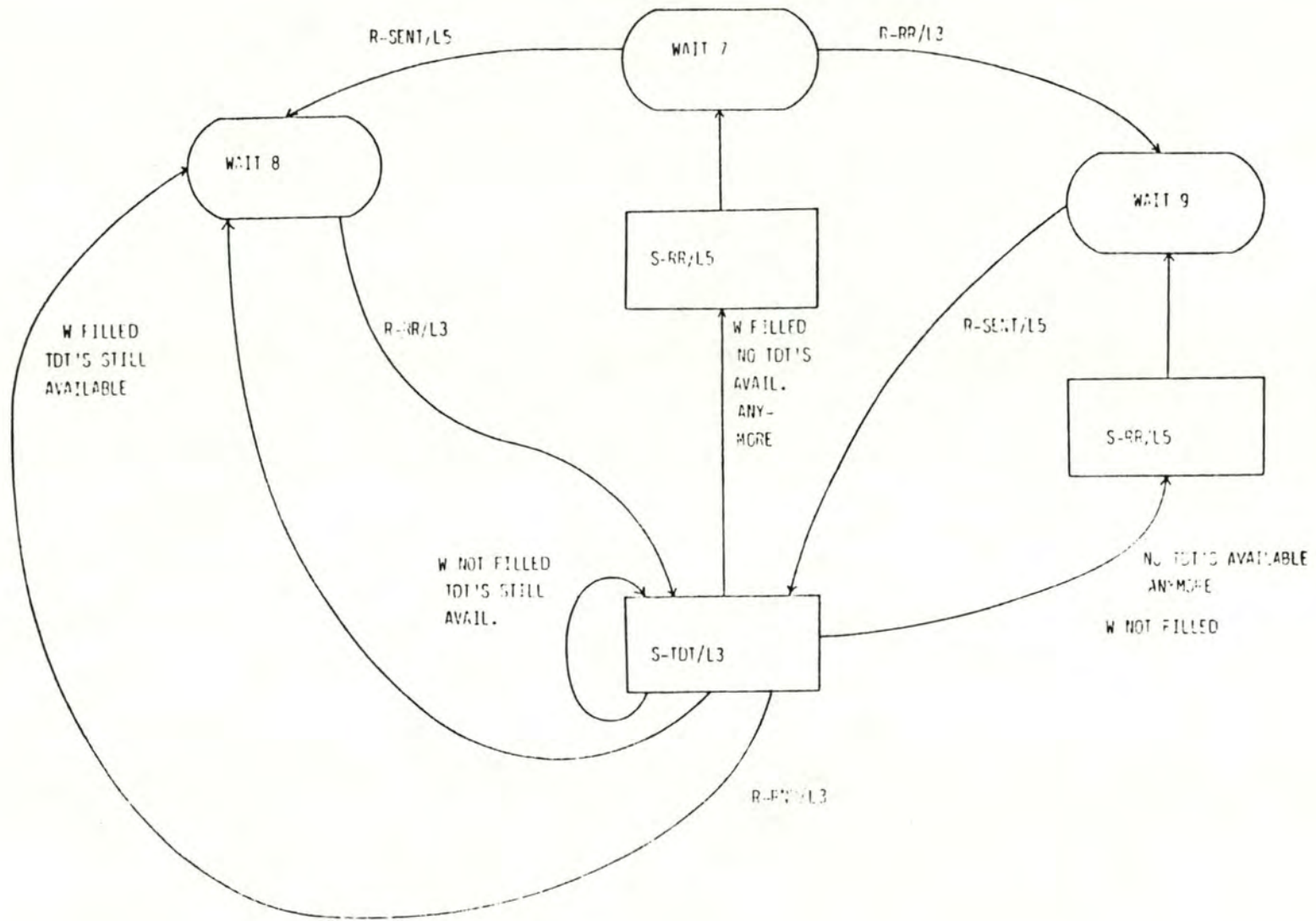


Fig. 51.5.5



Fig. 5.15.6



## DATA TRANSFER : RECEPTION

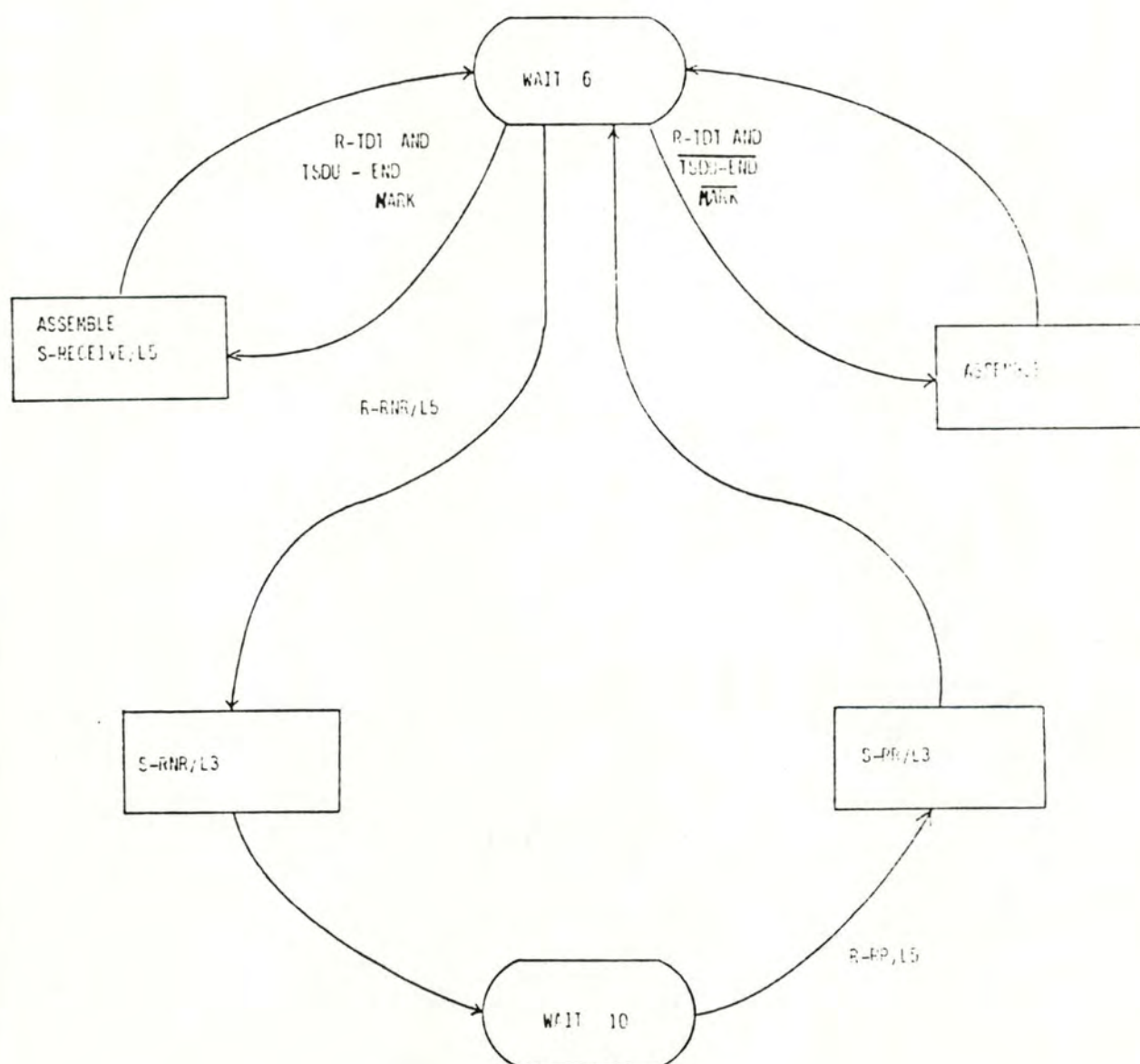


Fig. 5.15.7

## 5.2 - NIVEAU SESSION

### 5.2.1 - FONCTION GENERALE

Le niveau session gère principalement les fonctions d'échange dans une association entre deux entités logiquement définies. Ceci implique, par exemple, l'identification des correspondants, des services ainsi que la négociation des capacités réciproques des terminaux (largeur de fenêtre...).

### 5.2.2 - FONCTIONS DETAILLEES DU PROTOCOLE

On trouvera à la figure 5.2.2.1 le récapitulatif des primitives commandes et réponses pour les échanges au niveau 5 de session.

On trouvera aux figures suivantes sous trois formes différentes les paramètres associés à ces primitives:

- fig. 5.2.2.2: l'ensemble des primitives et la mise en oeuvre de celles-ci dans une séquence exemplative.
- fig. 5.2.2.3: les fonctions détaillées des primitives avec leurs paramètres, accompagnées de la mention "obligatoire" ou "non obligatoire".

On trouvera à la figure 5.2.2.4 le graphe d'états et des transitions pour l'appelant et l'appelé, destinés au contrôle du séquençement.

On trouve enfin en annexe C :

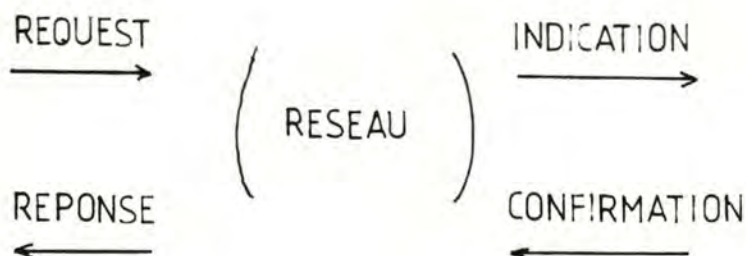
- les graphes précédents des états et des transitions, mais présentés sous forme d'organigramme et de tables séquentielles, plus utilisables à la programmation.



- un tableau comparatif des primitives:
  - du CCITT
  - d'une sélection effectuée par Meyer (cfr mémoire) dans ECMA
  - celles qui sont nécessaires lorsque la transaction est de type conversationnel.

Une remarque est à formuler au sujet de ces deux types de primitives ECMA - Meyer:

Les primitives que nous proposons ont été sélectionnées parmi celles déjà choisies. Ces dernières, en effet, étaient prévues pour être transmises dans un réseau où l'on distingue les messages entrants et sortants.



- a - Pour les primitives retenues, nous notons qu'il y a peu de discordance entre elles et les 5 couples de primitives fondamentales du CCITT.
- b - Les primitives supplémentaires présentent un intérêt car elles permettent:
  - d'initialiser et de désinitialiser les liaisons après tests et contrôles par S/R ACTIVATE.
  - l'envoi et la réception de messages urgents par S/R EXPEDITED.
  - le contrôle par l'appelant d'une demande de libération par l'appelé S (NOT/-) FINISHED.

Notons enfin que l'utilisation du "jeton" est une technique élégante pour résoudre certains problèmes, par exemple:

- les jetons DATA (S/R TOKEN GIVE, S/R TOKEN RECEIVE) permettent de modifier le sens de la transmission. Cette possibilité de prendre ou de donner la main à une station quelconque assure du même coup le contrôle des échanges. Notons enfin que la passation du pouvoir de contrôle est prévue par le CCITT avec les primitives CSCC et RSCCP.
- le jeton TERMINATE qui, à la réception d'un S-RELEASE, permet au récepteur:
  - de négocier le relâchement par un accord (S-FINISHED OK) ou par un refus (S-NOT FINISHED).
  - de ne pas négocier et de renvoyer uniquement l'accord (S-FINISHED OK).

## SESSION COMMANDS AND RESPONSES

COMMAND	RESPONSE	ABBREVIATION
SESSION ESTABLISHMENT AND CLEARING		
COMMAND SESSION START	RESPONSE SESSION START POSITIVE	CSS RSSP
	RESPONSE SESSION START NEGATIVE	RSSN
COMMAND SESSION END	RESPONSE SESSION END POSITIVE	CSE RSEP
COMMAND SESSION ABORT	RESPONSE SESSION ABORT POSITIVE	CSA RSAP
INFORMATION TRANSFER		
COMMAND SESSION USER INFORMATION	RESPONSE SESSION USER INFORMATION	CSUI RSUI
SESSION MANAGEMENT		
COMMAND SESSION CHANGE CONTROL	RESPONSE SESSION CHANGE CONTROL POSITIVE	CSCC RSCCP

Fig. 52.21



## ECHANGE

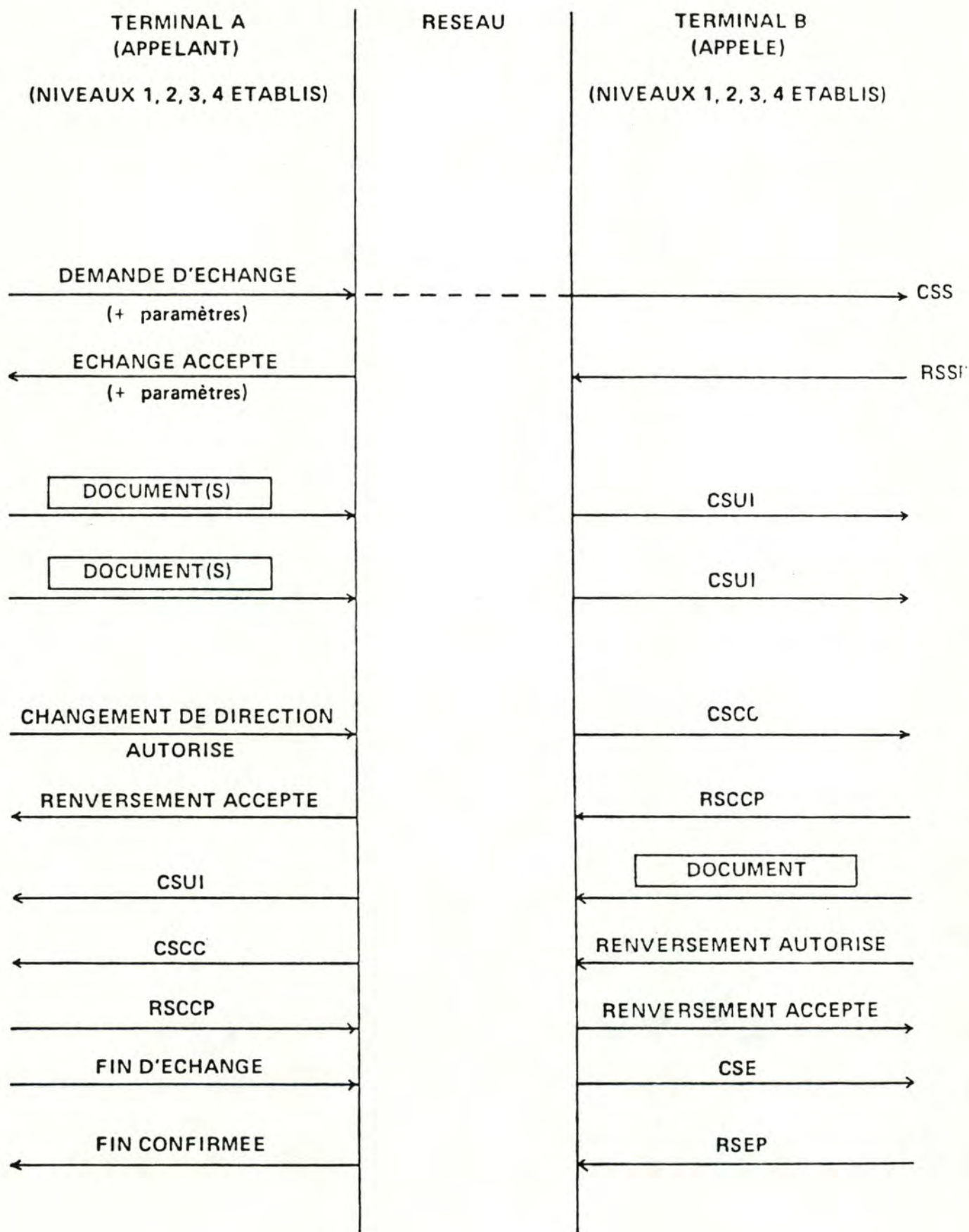


Fig. 5.222

## PGIs AND PLs FOR SESSION ELEMENTS OF PROCEDURE

SESSION COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY
CSS	SESSION REFERENCE	M	TERMINAL IDENTIFICA- TION OF THE CALLED TERMINAL	M
			DATE AND TIME	M
			ADDITIONAL SESSION REFERENCE NUMBER	NM
	NONBASIC SESSION CAPABILITY	NM	WINDOW SIZE	NM
			MISCELLANEOUS SESSION CAPABILITIES	NM
	NONBASIC TERMINAL CAPABILITIES	NM	CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANEOUS TERMI- NAL CAPABILITIES	NM
	PRIVATE USE	NM	—	
			SERVICE IDENTIFIER	M
CSE			SESSION TERMINATION	NM
CSA			SESSION TERMINATION	M
CSCC				
CSUI				

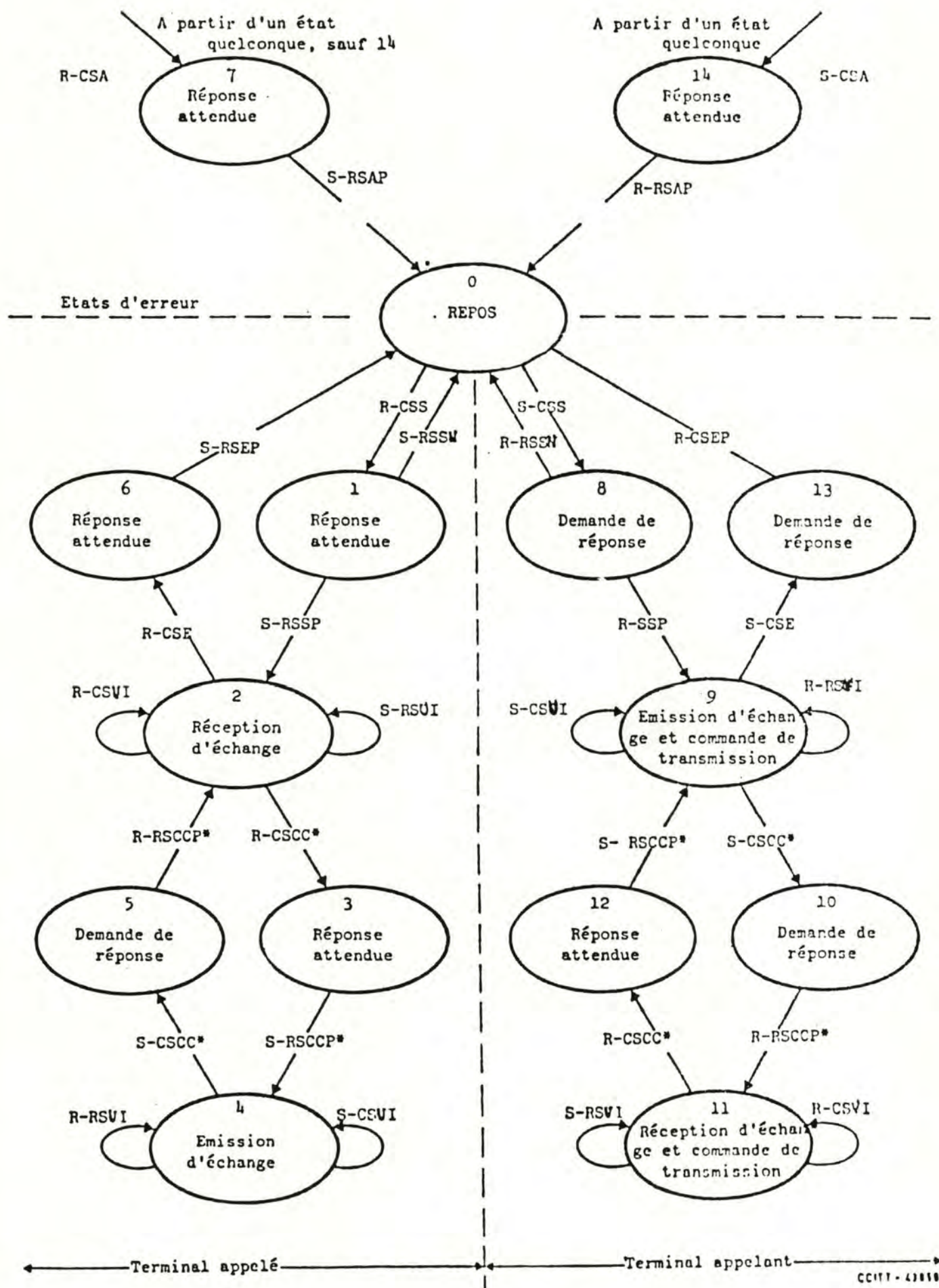
Fig. 5.2.2.3



SESSION COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY
RSSP	SESSION REFERENCE	M	TERMINAL IDENTIFICA- TION OF THE CALLED TERMINAL	M
			DATE AND TIME	M
			ADDITIONAL SESSION REFERENCE NUMBER	NM
	NONBASIC SESSION CAPABILITIES	NM	WINDOW SIZE	NM
			MISCELLANEOUS SES- SION CAPABILITIES	NM
	NONBASIC TERMINAL CAPABILITIES	NM	GRAPHIC CHARACTER SETS	NM
			CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANEOUS TER- MINAL CAPABILITIES	NM
			SERVICE IDENTIFIER	M
	PRIVATE USE	NM	—	—
			REQUEST SESSION FUNCTIONS	NM
RSSN	PRIVATE USE			NM
	FOR FURTHER STUDY	--	—	—
RSEP				
RSAP				
RSCCP				
RSUI			REQUEST SESSION FUNCTIONS	NM

Fig. 5.22.3 (suite)





\* Les commandes de contrôle de changement marquées d'un astérisque ne s'appliquent pas au mode d'échange OWC.

Fig. 5.2.24

## 5.3 - NIVEAU PRÉSENTATION

### 5.3.1 - IMPORTANCE DE CETTE COUCHE

Tout téléinformaticien est conscient que le format de présentation de documents sur des "lay-out" de terminaux différents pose parfois des problèmes aux solutions fort dispendieuses. Le formatage de ces terminaux est en effet fonction de leur technologie (électronique, micro-électronique...).

Toutes tentatives de classification de ces fonctions, selon les caractéristiques de ces terminaux, nécessitent la révision continuelle de ces classifications par modifications, suppression et création de paramètres.

Ceci explique que toutes normalisations ou banalisations réelles ou virtuelles restent sans effet, malgré la persévérance et la compétence de certains chercheurs. Beaucoup de constructeurs n'ont ni motivation, ni intérêt pour que les discordances soient pour le moins atténuées; le niveau de présentation des architectures SNA, DNA, CNA, DSA, n'apporte que peu de choses à la recherche de la normalisation des terminaux. Dès lors, les utilisateurs et les responsables des réseaux, devant la nécessité de rendre ceux-ci compatibles, ont dû trouver des moyens de banalisation des messages échangés.

Reprenons succinctement quels sont les problèmes soulevés par l'hétérogénéité des hôtes-serveurs et des terminaux.



Les réseaux d'ordinateurs offrent le moyen de mettre en relation deux processus. Un exemple important est la coopération entre l'opérateur d'un terminal et une application supportée par un serveur du réseau.

Chacun sait que la grande hétérogénéité caractérisant le marché des terminaux nécessite un effort d'adaptation spécifique, chaque fois qu'on désire offrir à un nouveau terminal la faculté d'accéder à une application donnée.

Dans le cadre d'un système informatique constitué par un serveur central unique auquel sont raccordés des terminaux locaux et distants, il est encore possible d'effectuer une adaptation pour chaque type de terminal raccordé au serveur central.

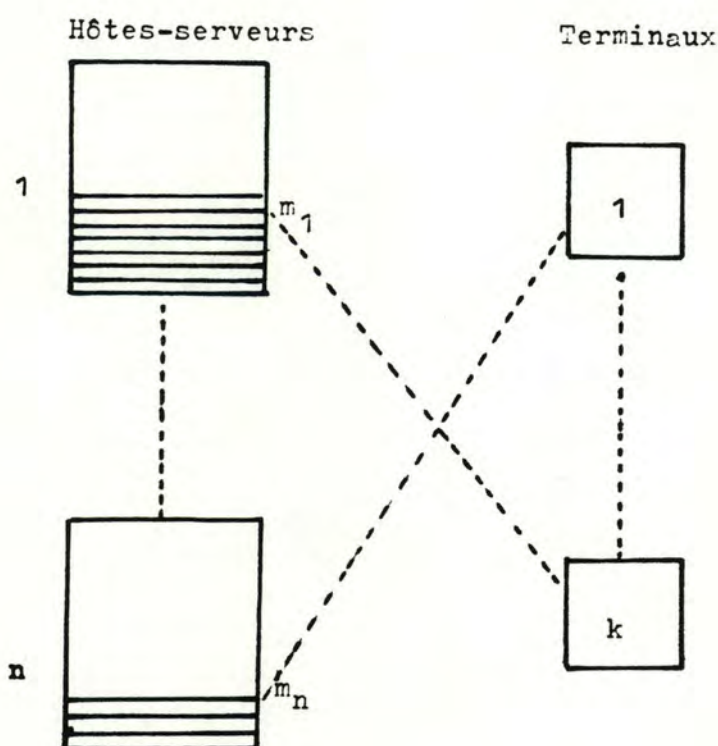
Par contre, dans l'optique d'un réseau d'ordinateurs, le problème de la diversité (des serveurs et des terminaux) se pose avec beaucoup plus d'acuité. Par conséquent, il est plus concevable de réaliser les multiples adaptations spécifiques permettant à chaque serveur de supporter directement chaque terminal raccordé au réseau.

Considérons à la limite un réseau maillé composé de  $n$  hôtes ou serveurs et de  $m_i$  programmes d'application distincts à la disposition de  $k$  terminaux différents. Pour assurer la compatibilité totale des échanges, il faudra :

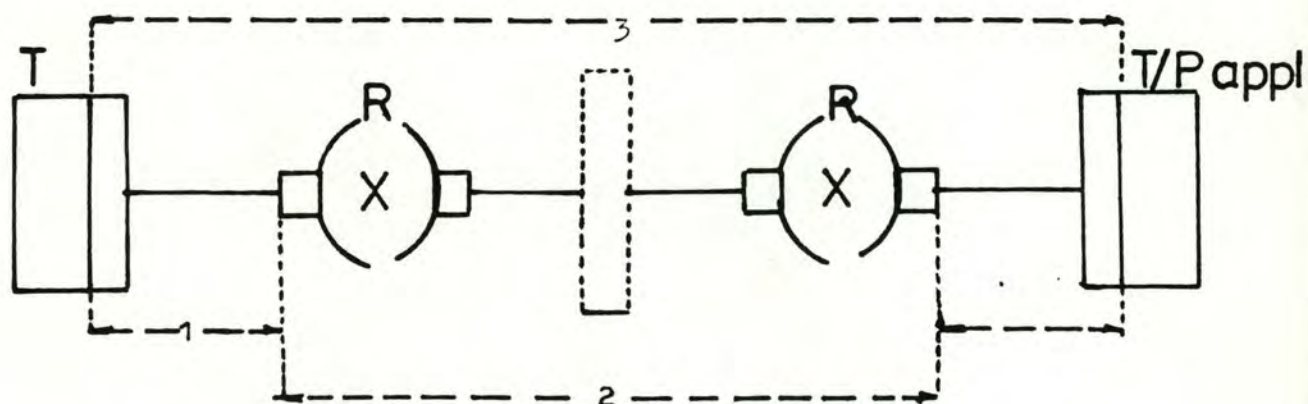
$$\sum_{i=1}^n m_i \cdot k = n \cdot m \cdot k \text{ logiciels différents.}$$



Si on normalise ou identifie les  $m_i$  programmes des ordinateurs et si on réalise la banalisation des terminaux, alors le nombre de logiciels différents peut se réduire à  $(n + m)$  ou  $m$  logiciels, selon que les ordinateurs distincts ou non doivent disposer d'un logiciel d'interface.



La normalisation qui suit les voies traditionnelles s'inspire de l'une ou l'autre des trois techniques illustrées figure 5.3.1, lorsqu'il s'agit des relations entre terminaux (T) et programme application, par le truchement d'un réseau.



1. Interfaces T-RES
2. Interfaces RES-RES
3. Interfaces log/mat T-T (cette solution est la plus fiable et la plus dispendieuse (chaque T est concerné) mais si le nombre des T normalisés est suffisant il y a une chute de prix.

Fig. 5.3.1

D'une manière générale et quelles que soient les liaisons, la traduction entre les commandes externes fort diverses et celles qui seront normalisées et interprétables par le receveur, peut se situer:

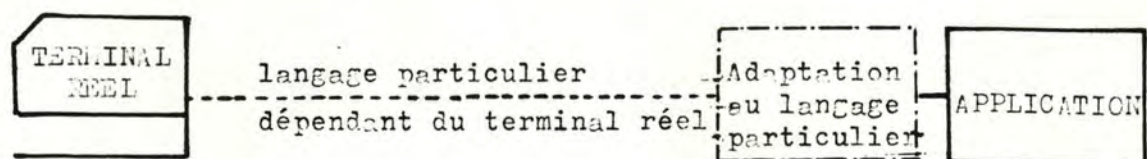
- dans le terminal lui-même de façon intégrée ou complémentaire.
- dans l'ordinateur de traitement ou de son frontal.
- dans des équipements d'informatiques à fonctions diverses (concentrateurs, interfaces...).
- dans des équipements de réseaux de transmission de données.



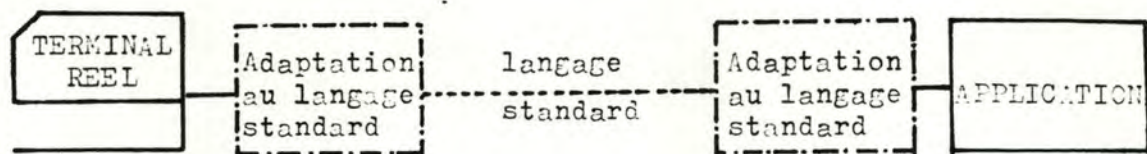
Une solution plus "élégante" et théoriquement plus valable consiste à définir un langage standard d'interaction entre un terminal et une application. Ce langage serait valable pour l'ensemble du réseau et serait obligatoirement utilisé lors de tout dialogue entre un terminal raccordé au réseau et une application disponible sur celui-ci. Cette solution implique évidemment qu'une adaptation aux caractéristiques du langage standard soit entreprise tant du côté du terminal que du côté de l'application. En d'autres termes, plutôt que de réaliser une adaptation spécifique à chaque terminal réel ce qui implique que le dialogue soit organisé à l'aide d'un langage qui reste influencé par les caractéristiques du terminal réel (voir fig. 5.3.2a), on préférera effectuer deux adaptations locales à un langage standard (voir fig. 5.3.2.b). Ce dernier étant défini à l'échelle du réseau, les adaptations locales (du terminal réel, d'une part, de l'application de l'autre) ne devront être effectuées qu'une seule fois. L'adaptation du terminal réel au langage standard définit un terminal abstrait que nous appellerons terminal virtuel (voir fig. 5.3.2c).

Reportons-nous à présent à la figure 5.3.2d. Cette figure fait apparaître la décomposition de chacun des modules locaux d'adaptation au langage standard (représentés à la figure 5.3.2c) en deux sous-modules fonctionnels dont l'un joue un rôle exclusivement local (sous-module émulateur ou sous-module interface selon le cas) tandis que l'autre participe au processus de dialogue de bout en bout (sous-modules station VTP  $\equiv$  Virtual Terminal Protocol).

Le module d'adaptation associé au terminal réel se décompose en un sous-module local dénommé émulateur et un sous-module de dialogue de bout en bout appelé station VTP.



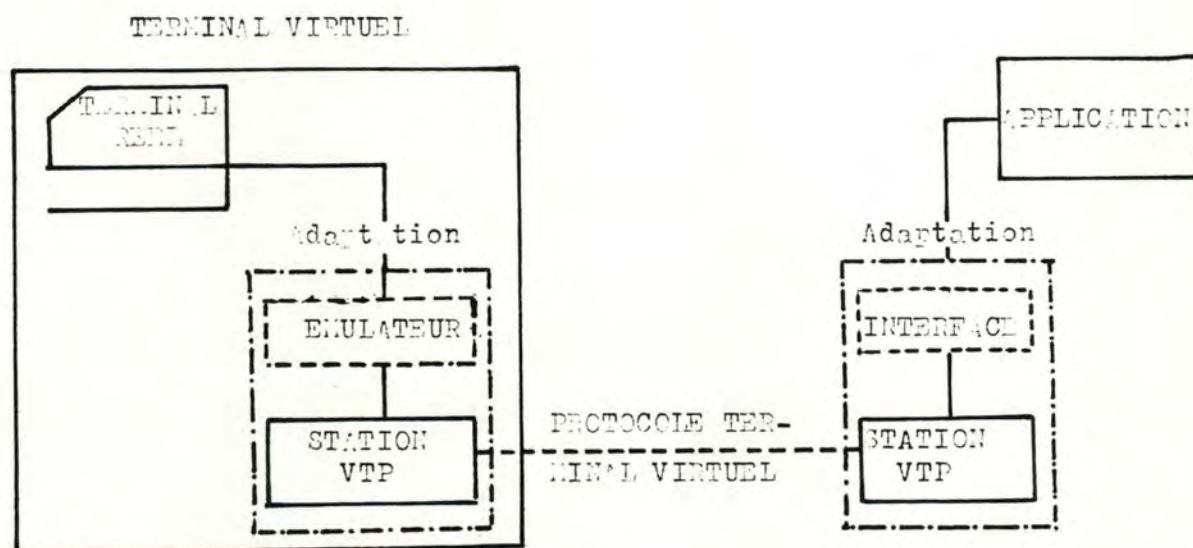
a



b



c



d

- Fig. 5.3.2 -



Nous reparlerons plus loin des propositions qui viennent d'être faites.

Le problème étant posé, nous allons cataloguer les solutions utilisées actuellement. Ainsi, lorsqu'on examine les techniques d'adaptation adoptées, on peut les récapituler en 5 classes pour lesquelles nous allons examiner les fonctions détaillées après avoir présenté les fonctions générales.

### 5.3.2 - FONCTIONS GENERALES

#### a - Négociation (L7)

- réaffirme les paramètres négociés au niveau 5 et les adapte à l'application concernée.
- négocie la taille mémoire (CDCL / RDCLP avec éventuellement un RDGR s'il y a discordance.

#### b - Fonctions (L6)

fournit les fonctions nécessaires pour interpréter les données reçues de L5 du point de vue alphabétique, format, syntaxe et parfois sémantique. Par exemple, en videotex, le codage des couleurs est interprété à ce niveau et les adaptations sont faites pour l'écran TV couleur (L7). Si l'on utilise une imprimante couleur, l'adaptation est différente pour un même codage ou interprétation.

- Fonctions normales

- CDS (début) ou CDC (reprise):  
confirmation, type document...
- CDUI: données
- CDE (RDEP): fin normale.
- CDR (RDRP): fin prématurée,  
erreur à l'émission.
- CDD (RDDP): fin prématurée.  
expédition responsable.

- Fonctions complémentaires

- CDRP: communication du dernier point  
de repère  
RDRPP: confirmation du dernier  
point de repère.  
RDRPN: discordance - reprise.
- CDRPR communication du point de  
reprise.  
RDRPR: autorise la reprise.
- CDPB: indique le numéro du point de  
repère (à chaque page).  
RDPBP: confirme le numéro du  
point de repère; si la  
saturation mémoire  
n'est pas atteinte.



### 5.3.3 - FONCTIONS DETAILLEES DES PROTOCOLES POSSIBLES

Plusieurs protocoles sont, en effet, possibles pour assurer la compatibilité entre "terminaux-programmes". Nous les examinerons chronologiquement dans l'ordre de leurs performances.

#### 5.3.3.1 - Classe 0

C'est le moyen le plus élémentaire qui consiste à mémoriser, dans le terminal (ou s'il y a grappe de terminaux dans le noeud programmé de ceux-ci), l'ensemble des paramètres du ou de chacun des terminaux.

Le programme inséré dans le terminal ou dans l'interface formatera le message incident en fonction des exigences de la présentation formulées par ces paramètres.

On pourrait par exemple, pour le teletex, sélectionner ceux indiqués dans la figure 5.3.3.1.1.

	Paramètres d'un format	valeur par défaut des paramètres.
P <sub>1</sub>	lignes par page	55
P <sub>2</sub>	caractères par ligne	77
P <sub>3</sub>	interligne	4,23 mm
P <sub>4</sub>	espacement entre caractères	2,54 mm
P <sub>5</sub>	position hot-zone	71
P <sub>6</sub>	longueur hot-zone	6
P <sub>7</sub>	tabulateur (on peut en définir au moins 8)	3 <sup>e</sup> carac.
P <sub>8</sub>	emplacement de la pagination (haut/bas de page)	0
P <sub>9</sub>	position de la pagination (position horizontale)	0
P <sub>10</sub>	debut du texte sur chaque page (nombre de lignes blanches avant le debut du texte)	5
P <sub>11</sub>	longueur du formulaire (papier continu)	0
P <sub>12</sub>	alimentation feuille/feuille automatique	oui
P <sub>13</sub>	mode d'utilisation des cassettes multiples	selon la technique
P <sub>14</sub>	jeu de caractères utilisés. De plus, en format libre et si l'imprimante autorise d'autres jeux de caractères, différentes options sont possibles	(cfr avis S.61)

- Fig. 5.3.3.1.1.-



### 5.3.3.2 - Classe I

Dans cette classe, on trouvera les moyens issus des codes spéciaux répertoriés dans l'alphabet IA5 du code ASCII soit:

- 1 - à utilisation directe. Ce code dont la grille est reproduite à la figure 5.3.3.2.1 est précisé en annexe D , du moins les fonctions des différents caractères en relation avec la présentation d'un message. Pour la transmission de textes, on inclut les caractères PLD et PLU pour fournir respectivement un interligne vers le bas et vers le haut (fig. 5.3.3.2.2)  
Ex.: A. PLD. 2 = A<sub>2</sub>
- 2 - à utilisation complémentaire du code ASCII:
  - pour les caractéristiques de présentation telles que:
    - PFS (tabulation horizontale /verticale)
    - SGR (caractère de présentation)
    - SHS (Espacement horizontal)
    - SVS (Espacement vertical)
  - pour les caractéristiques d'extension telles que  
ESC. CSI P<sub>1</sub>, P<sub>2</sub>...
  - pour les caractères spécifiques aux terminaux.

Pour plus de détails concernant ces caractères, il y a lieu de se reporter à l'annexe D.

## ALPHABET INTERNATIONAL. N° 5

					b <sub>7</sub>	0	0	0	0	1	1	1	1				
					b <sub>6</sub>	0	0	1	1	0	0	1	1				
					b <sub>5</sub>	0	1	0	1	0	1	0	1				
						0	1	2	3	4	5	6	7				
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		0	0	0	0	0	NUL	TC <sub>7</sub> (DLE)	SP	0	@	P	`	p
0	0	0	1	1	0	0	0	1	1	TC <sub>1</sub> (SOH)	DC <sub>1</sub>	!	1	A	Q	a	q
0	0	1	0	2	0	0	1	0	1	TC <sub>2</sub> (STX)	DC <sub>2</sub>	"	2	B	R	b	r
0	0	1	1	3	0	0	1	1	0	TC <sub>3</sub> (ETX)	DC <sub>3</sub>	#	3	C	S	c	s
0	1	0	0	4	0	1	0	0	1	TC <sub>4</sub> (EOT)	DC <sub>4</sub>	□	4	D	T	d	t
0	1	0	1	5	0	1	0	1	0	TC <sub>5</sub> (ENQ)	TC <sub>8</sub> (NAK)	%	5	E	U	e	u
0	1	1	0	6	0	1	1	0	1	TC <sub>6</sub> (ACK)	TC <sub>9</sub> (SYN)	&	6	F	V	f	v
0	1	1	1	7	0	1	1	1	0	BEL	TC <sub>10</sub> (ETB)	'	7	G	W	g	w
1	0	0	0	8	1	0	0	0	0	FE <sub>0</sub> (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	1	0	0	1	0	FE <sub>1</sub> (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	1	0	1	0	0	FE <sub>2</sub> (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	11	1	0	1	1	0	FE <sub>3</sub> (VT)	ESC	+	;	K	[	k	{
1	1	0	0	12	1	1	0	0	0	FE <sub>4</sub> (FF)	IS <sub>4</sub> (FS)	,	<	L	\	l	
1	1	0	1	13	1	1	0	1	0	FE <sub>5</sub> (CR)	IS <sub>3</sub> (GS)	-	=	M	]	m	}
1	1	1	0	14	1	1	1	0	0	SO	IS <sub>2</sub> (RS)	.	>	N	^	n	-
1	1	1	1	15	1	1	1	1	0	SI	IS <sub>1</sub> (US)	/	?	O	_	o	DEL

Fig. 5.3.2.1



# ASCII ETENDU

					e	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
--	--	--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fig. 5.3.3.2.2.

### 5.3.3.3.- Classe II

On groupera dans cette classe les moyens initialement proposés par l'ISO et concrétisés par le CCITT.

Comme pour le niveau session:

- a - Nous trouvons aux figures 5.3.3.3. 1 à 3, respectivement:
  - l'ensemble des primitives commandes et réponses pour les échanges du niveau 6 de présentation (cette liste se retrouve également en annexe E).
  - la correspondance de ces primitives au niveau des sous-séquences du dialogue dans un cas de transmission sans anomalies et un cas avec anomalies.
- b - Les figures 5.3.3.3.4 à 6 donnent la liste des paramètres associés aux primitives du niveau 6. Les fonctions détaillées de ces paramètres ainsi que les actions qu'ils exercent pourront se trouver dans l'annexe E.
- c - Tout ce qui concerne le séquençement des opérations, c'est-à-dire l'ensemble des graphes relatifs au niveau 6 en prenant en considération les primitives propres à cette couche, se trouve aux figures 5.3.3.3 7/8.



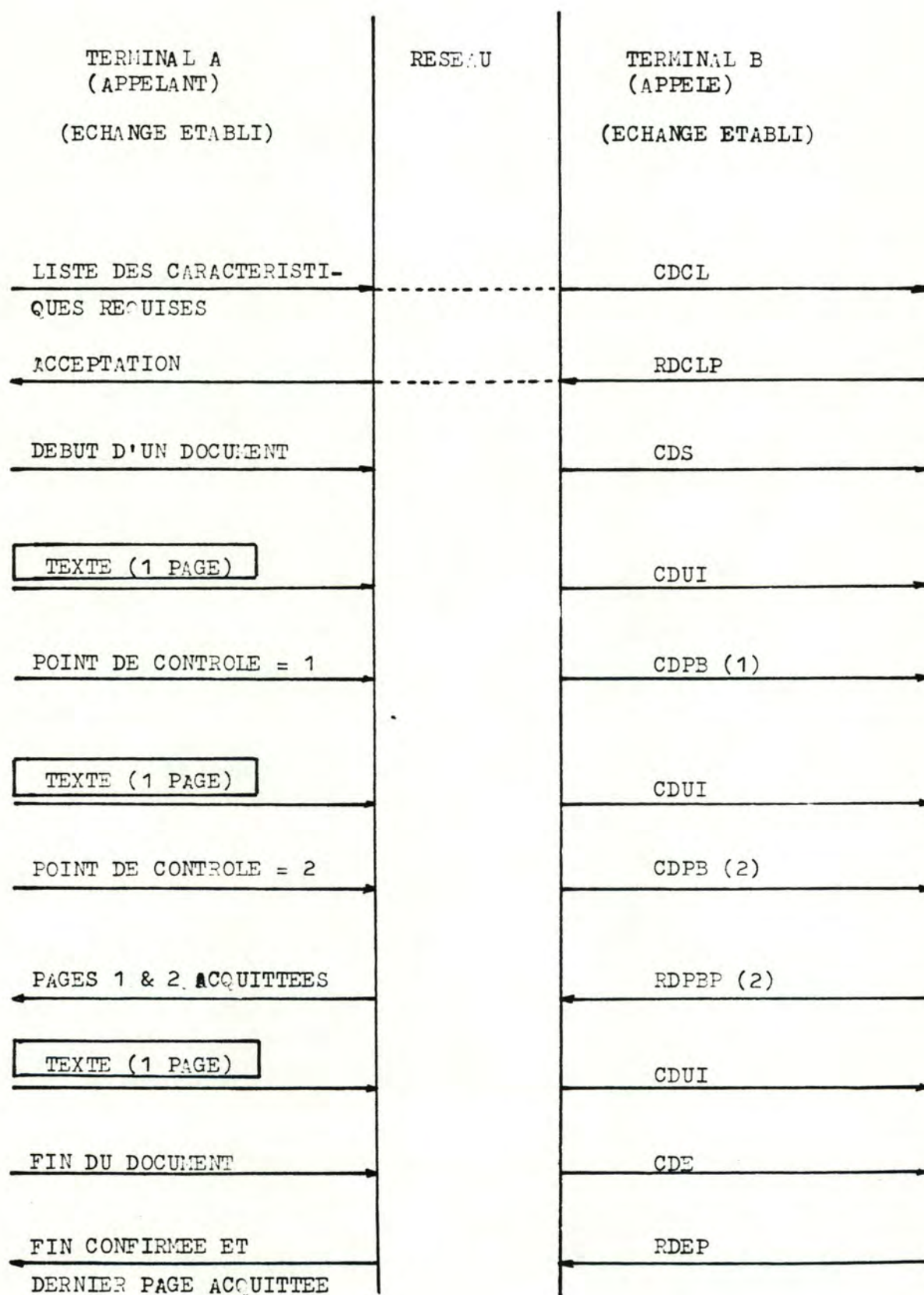
Les graphes des états et de transitions destinés au contrôle des séquences du côté appelant comme du côté appelé sont donnés à ces figures.

(La description générale et les règles de fonctionnement applicables aux diagrammes des états sont donnés(en annexe A). L'enchaînement de ces séquences est schématisé par deux tables séquentielles se trouvant en annexe E.Ces tables pourront faciliter une programmation future de la couche.

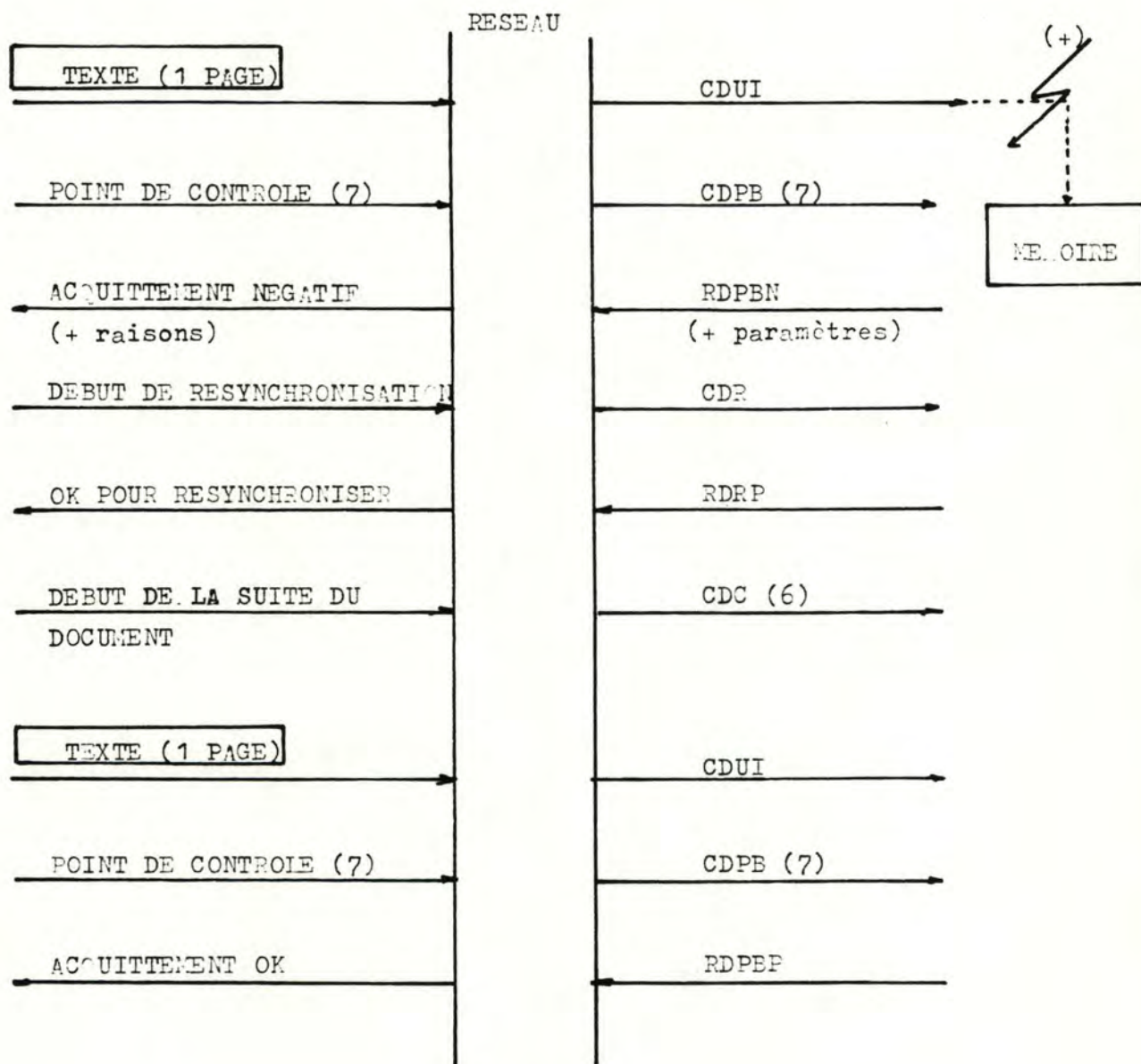
COMMAND	RESPONSE	ABBREVIATION
DOCUMENT CONTROL		
COMMAND DOCUMENT START		CDS
COMMAND DOCUMENT CONTINUE		CDC
COMMAND DOCUMENT CAPABILITY LIST		CDCL
	RESPONSE DOCUMENT CA- PABILITY LIST POSITIVE	RDCLP
COMMAND DOCUMENT END		CDE
	RESPONSE DOCUMENT END POSITIVE	RDEP
COMMAND DOCUMENT DISCARD		CDD
	RESPONSE DOCUMENT DISCARD POSITIVE	RDDP
COMMAND DOCUMENT RESYNCHRONIZE		CDR
	RESPONSE DOCUMENT RESYNCHRONIZE POSITIVE	RDRP
INFORMATION TRANSFER		
COMMAND DOCUMENT INFORMATION		CDUI
ERROR RECOVERY		
	RESPONSE DOCUMENT GENERAL REJECT	RDGR
COMMAND DOCUMENT PAGE BOUNDARY		CDPE
	RESPONSE DOCUMENT PAGE BOUNDARY POSITIVE	RDPBP
	RESPONSE DOCUMENT PAGE BOUNDARY NEGATIVE	RDPBN

DOCUMENT COMMANDS AND RESPONSES





TRANSMISSION SANS ANOMALIES



(+) Causes éventuelles de défaillance

- mémoire saturée
- rupture de séquence
- erreur locale du terminal

TRANSMISSION AVEC ANOMALIES



DOCUMENT COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY
CDS			DOCUMENT REFERENCE NUMBER	M
			SERVICE INTERWORKING IDENTIFIER	NM
			DOCUMENT TYPE IDENTIFIER	NM
	NONBASIC TERMINAL CAPABILITIES	NM	GRAPHIC CHARACTER SETS	NM
			CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANEOUS TERMI- NAL CAPABILITIES	NM
	PRIVATE USE	NM	—	—
CDC	DOCUMENT LINKING    (ONLY REQUIRED IF LINKING IN A NEW SESSION)	M	DOCUMENT REFERENCE NUMBER	N
			CHECKPOINT REFERENCE NUMBER	M
			TERMINAL IDENTIFIER OF THE CALLING TERMINAL	M
			TERMINAL IDENTIFIER OF THE CALLED TERMINAL	M
			DATE AND TIME	M
			ADDITIONAL SESSION REFERENCE NUMBER	NM
			SERVICE INTERWORKING IDENTIFIER	NM
			DOCUMENT TYPE IDENTIFIER	NM
			DOCUMENT REFERENCE (CURRENT SESSION)	M
			OTHER PARAMETERS OF CDS	NM
	PRIVATE USE	NM	—	—

Fig. 5.3.3.4



DOCUMENT COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- TORY
CDE			CHECKPOINT REFEREN- CE NUMBER	M
CDR			REASON (DOCUMENT)	NM
CDD			REASON (DOCUMENT)	NM
CDPB			CHECKPOINT REFEREN- CE NUMBER	M
CDCL			STORAGE CAPACITY NEGOTIATION	NM
	NONBASIC TERMINAL CAPABILITIES	NM	GRAPHIC CHARACTER SETS	NM
			CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANEOUS TER- MINAL CAPABILITIES	NM
	PRIVATE USE	NM	—	
CDUI			—	—
CDRP			RECOVERY POINT REFERENCE NUMBER	M
CDRPR			RECOVERY POINT REFERENCE NUMBER	M
RDEP			CHECKPOINT REFEREN- CE NUMBER	M
RDRP			—	—
RDDP			—	—
RDPBP			CHECKPOINT REFEREN- CE NUMBER	M
			RECEIVING ABILITY JEOPARDIZED	M
RDPBN			REASON (DOCUMENT)	M

Fig. 5.3335



DOCUMENT COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY
RDCLP			ACCEPTANCE OF CDCL PARAMETERS	NM
			STORAGE CAPACITY NEGOTIATION	NM
	NONBASIC TERMINAL CAPABILITIES	NM	GRAPHIC CHARACTER SETS	NM
			CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANEOUS TERMI- NAL CAPABILITIES	NM
	PRIVATE USE		—	—
RDGR			REFLECT PARAMETER VALUES	M
RDRPN			RECOVERY POINT RE- REFERENCE NUMBER	M
RDRPR			RECOVERY POINT RE- REFERENCE NUMBER	M

Fig. 5.33.36

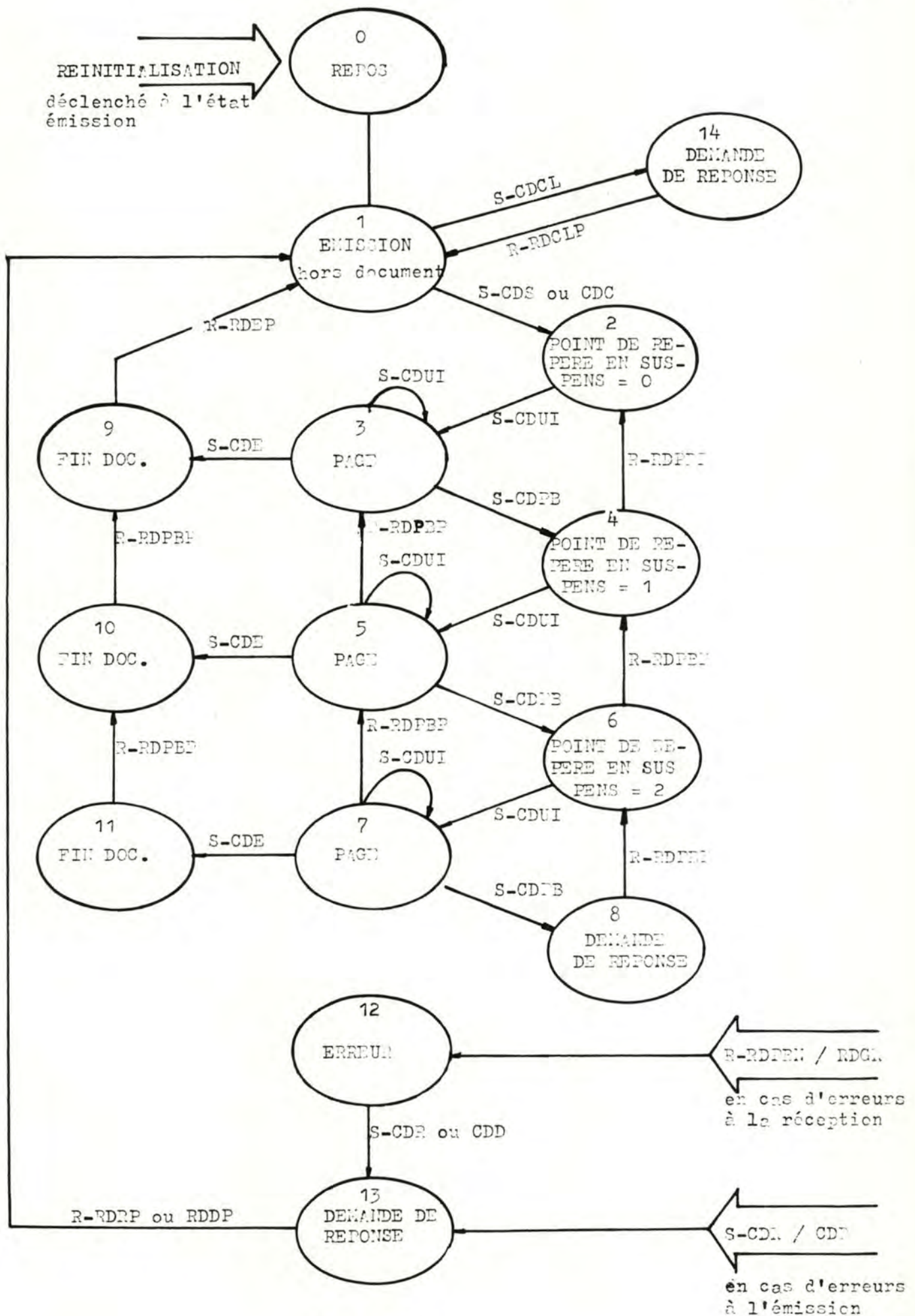


Fig. 5.3.3.3.7



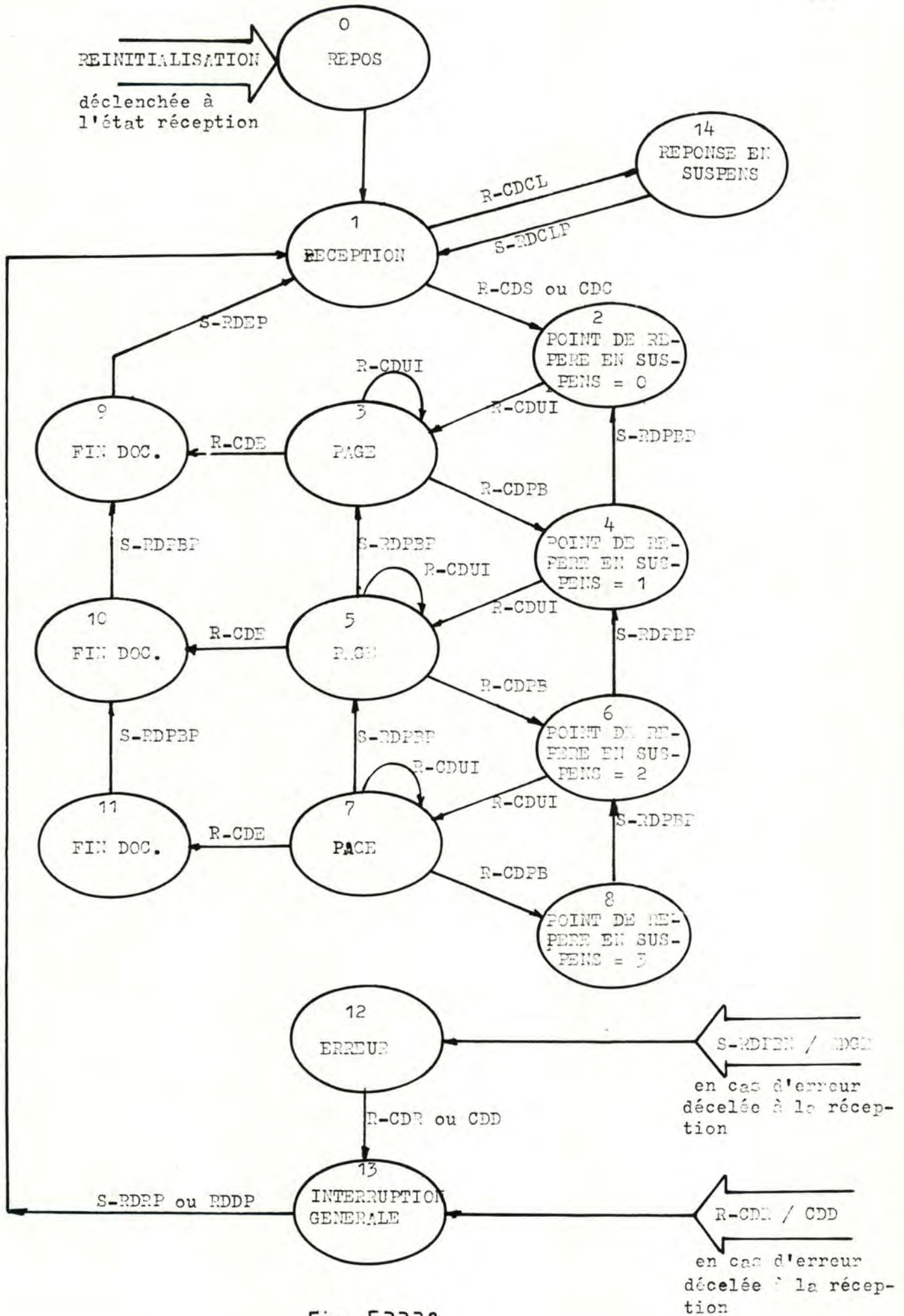


Fig. 53338

#### 5.3.3.4 - Classe III

Il existe pour la transmission asynchrone, (avec un prolongement futur pour la transmission synchrone), dans les réseaux de commutation par paquet, un choix (limité physiquement à 256) de services complémentaires susceptibles d'aider l'utilisateur non seulement dans la présentation mais également dans l'exploitation des messages.

Les terminaux asynchrones sont raccordés au RCP via un module d'entrée appelé ADP (Assembleur - Désassembleur de Paquets).

Le fonctionnement de l'ADP correspond à un certain nombre de fonctions nécessaires pour assurer un service minimum aux usagers. Il est appelé "mode de base". D'autres fonctions ou d'autres manières d'exécuter les fonctions de base sont offertes aux usagers: "les services complémentaires". A un instant donné, le comportement de l'ADP est déterminé par un ensemble de paramètres propres à un terminal donné. La valeur de chaque paramètre indique si la fonction correspondante est opérationnelle ou les caractéristiques de la fonction.

Chaque paramètre est référencé par une valeur décimale entre 0 et 255, permettant ainsi au terminal d'accéder à un paramètre donné pour le modifier.

A l'établissement d'une communication, les paramètres ont pour valeur initiale celle qui correspond au mode de base. Les valeurs sont soulignées dans la suite de la section.



La liste de la figure 5.3.3.4 1 et 2 est limitée aux 18 services complémentaires les plus couramment implantés dans les réseaux de transmission de données. Ils sont loin d'être exhaustifs ( $\leq 256$ ).

L'ADP ou tout module équivalent, dispose d'un protocole  $X_{28}$  (fig. 5.3.3.4. 3) qui permet un certain nombre d'opérations (lecture, mise en oeuvre, modification de paramètres...) avant ou pendant le transfert d'informations.

# SERVICES COMPLEMENTAIRES (asynchrone)

Référence	Valeurs	Signification
1	0: Interdit <u>1</u> : Possible	Retour en phase de commande sur détection de DLE + CR (autorise LIB, INT, RESET, STAT...)
2	0: Pas d'écho <u>1</u> : écho	Echo age des caractères reçus par l'ADP
3	0: Non 2: R C <u>126</u> : DLE, ETX	Sélection signal envoi données + caract.: 126 (aussi: délai, paquet plein)
4	0, 1, ... <u>20</u> ..255	Délai d'envoi de données (1/20 sec. )
5	0: non <u>1</u> : <u>oui</u>	XON: ADP accélère T PAD → T XOFF: ADP freine T
6	0: non <u>1</u> : <u>oui</u>	Transmission des indications au terminal par l'ADP
7	0: rien (BREAK) 1: INT <u>2</u> : <u>RESET</u> 3: INT + RUPT 4: COMMANDE	Procédure de l'ADP venant du terminal
8	0: Normal 1: arrêt	Remise des données au terminal
9	<u>0</u> <u>1</u> : 7	Nombre de NUL'S de bourrage après RC
10	<u>0</u> 1, ..., 255	Remplissage après RC avec éventuellement pliage (dépassement ligne physique)

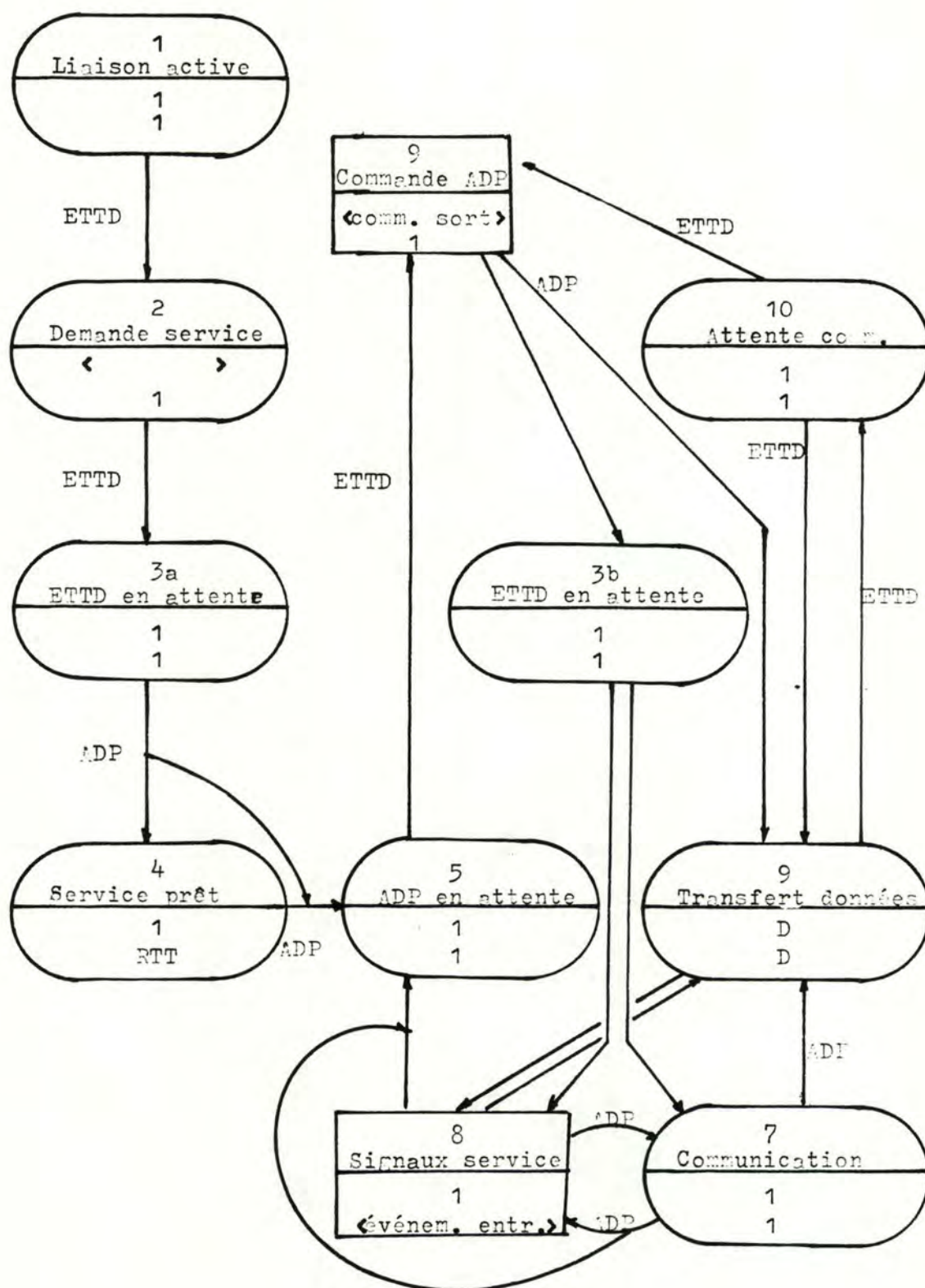
—: par défaut



# SERVICES COMPLEMENTAIRES (asynchrone)

Référence	Valeurs	Signification
11	0, 2, 8	Vitesse linéaire de la ligne 50 b/s ... 1200 b/s ... 64 kb/s
12	0 1	$T \longrightarrow$ ADP $X_{ON}$ $X_{OFF}$
13	0 1 2 4 + combinais.	$\overline{LF}$ LF après chaque RC du Terminal LF après chaque RC du Terminal LF après chaque RC envoyé comme écho au Terminal
14	0 1-7	Pas de remplissage après LF Nombre de caractères après LF
15	0 1	Non recours à l'édition Recours à l'édition (param. 16 à 18)
16	0 - 127	Effacement de caractères ( $IA_5$ )
17	0 - 127	Effacement de ligne au niveau du caractère ( $IA_5$ )
18	0 - 127	Affichage de ligne au niveau du caractère ( $I_5$ )

- Fig. 5.3.3.4.2.-



X.28



#### 5.3.3.5 - Classe IV

En introduction (§ 5.3.1), nous avons signalé que la solution la plus logique, la plus élégante et la plus performante au problème de la standardisation universelle de la présentation, était l'utilisation du PAV ou protocole d'appareil virtuel ( $\equiv$  VTP).

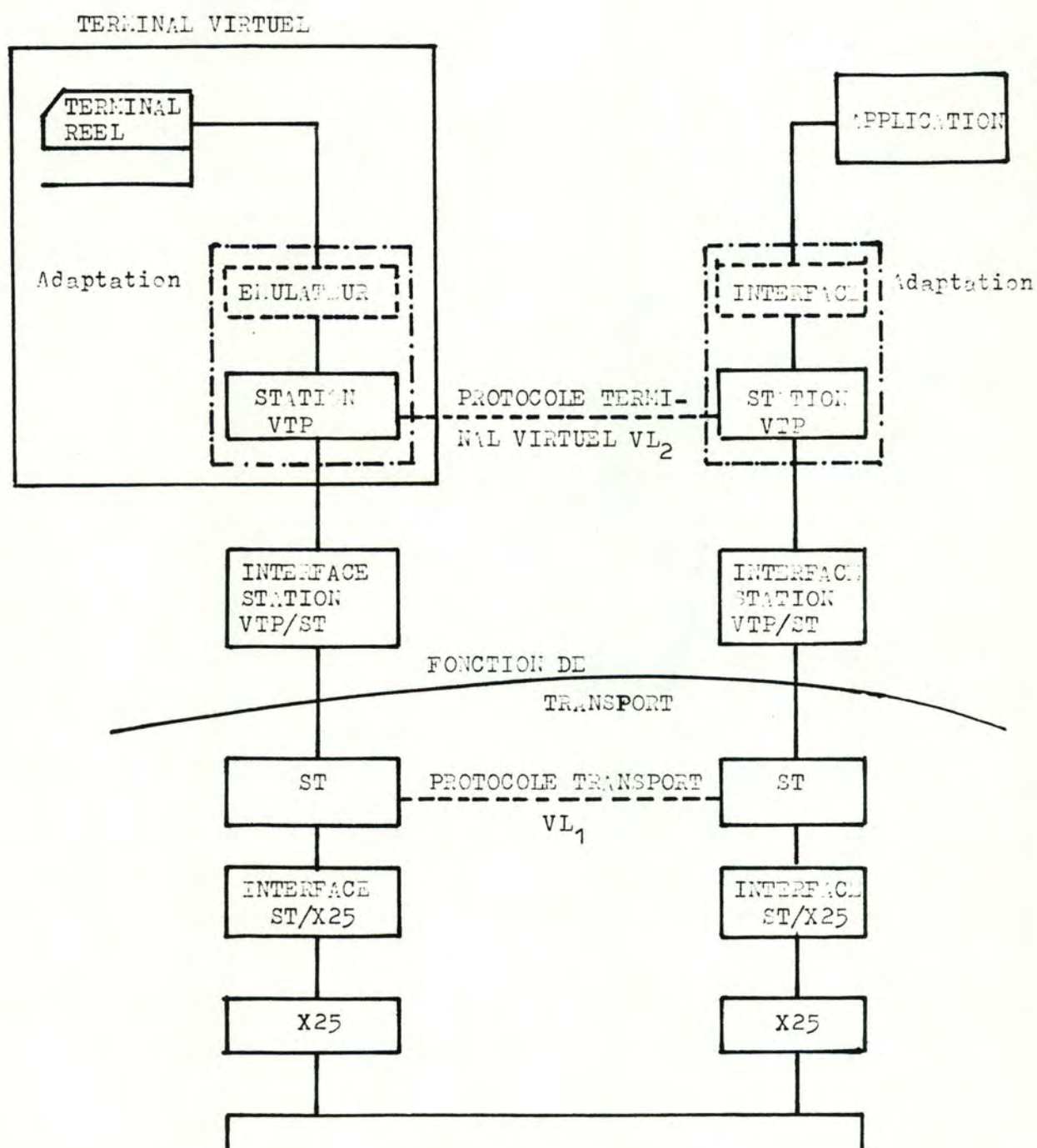
Nous avons dit également qu'étant donné la mouvance de la technologie et le peu d'intérêt apporté par les constructeurs à la normalisation, cette solution n'était appliquée que par peu d'utilisateurs et assez partiellement.

Nous formulerons, ci-après les grands principes de cette technique. Reprenons le schéma-bloc de la figure 5.3.2 sous la forme de celui de la figure 5.3.3.5.1 .

L'émulateur gère le terminal réel en traduisant les primitives du terminal virtuel en primitives interprétables par le terminal réel (et vice-versa). Sa fonction reste purement locale.

La station VTP organise l'échange des primitives du terminal virtuel entre entités distantes. Elle dialogue pour ce faire avec la station VTP distante.

Du côté de l'application, l'interface joue un rôle de traducteur comparable à celui de l'émulateur du côté du terminal réel.



TERMINAL, APPLICATION ET FONCTION DE TRANSPORT



Le but est double:

- a - définir le terminal abstrait ou virtuel dont nous venons de dégager le concept.
- b - définir le langage d'interaction standard ou protocole terminal virtuel (VTP) régissant le dialogue entre les deux stations VTP distantes sur la liaison virtuelle  $VL_2$  (fig. 5.3.2d).

La figure 5.3.3.5.1 fait apparaître le protocole terminal virtuel comme un protocole de niveau utilisateur se superposant au protocole de transport. En d'autres termes, la liaison virtuelle  $VL_2$  est implémentée à l'aide de la liaison virtuelle  $VL_1$  entre stations de transport (ST). Les services offerts par le protocole de transport (niveau 4) sont limités à:

- a - l'échange de lettres, sur lesquelles sont assurés les services suivants: séquençement, contrôle d'erreurs sur l'arrivée et sur le contenu.
- b - l'échange de télégrammes, véhiculés de façon prioritaire mais pour lesquels aucun des services offerts sur les lettres n'est assuré.

Quant à la liaison virtuelle  $VL_1$ , elle est implémentée par le réseau de communication (RCP = réseau à commutation de paquets) prolongé par l'interface standard X25 du CCITT.

Le protocole terminal virtuel utilise blocs et télégrammes, tandis que le protocole de transport utilise lettres et télégrammes (notion d'indépendance des niveaux de protocole). Une interface station VTP/ST est dès lors nécessaire.

La première démarche dans l'étude d'un appareil virtuel est de dresser la liste des caractéristiques des terminaux.

On trouvera en annexe F la liste des catégories des paramètres des terminaux, sans entrer dans les détails de ceux-ci.

Evidemment, ceci nécessite au préalable une négociation relative au choix des paramètres adoptés de part et d'autre. Cette négociation pourra également se faire en cours d'échange, ce qui implique au moins un protocole et des primitives paramétrisées.

Les grandes lignes de ce protocole d'appareil virtuel et des primitives utilisées sont données respectivement dans l'annexe F.



Au vu de ce paragraphe relatif aux appareils virtuels de présentation, on constate que l'on a substitué le problème de normalisation du PAV au problème de la banalisation ou de l'émulation des terminaux.

Cette raison s'ajoutant à la prolifération des types de terminaux, explique que cette solution a des difficultés de s'imposer.

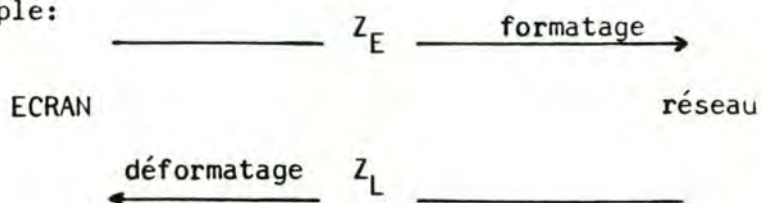
## 5.4 - APPLICATION

Le niveau 7 est celui laissé à la liberté de l'utilisateur, ce qui veut dire que les programmes de cette couche sont de nature, de complexité et d'exigences fort inégales. Nous avons développé dans ce chapitre et dans le détail, un ensemble complet à couche de transport (4), de session (5) et de présentation (6) pour les échanges de textes normalisés. Si l'application porte sur des transactions toutes différentes comme le contrôle de processus, le transactionnel, le conversationnel... et non sur la qualité des documents, les principes proposés resteront valables. Toutefois, on assistera comme nous le verrons au chapitre 7 à la mutilation de primitives, de paramètres, de fonctions de la couche session et même de couches. Inversement, il y aura la création de primitives et de fonctions. Dès lors, la normalisation se limitera à l'ensemble fermé des usagers qui ont adopté ces nouveaux principes.

Dans tous les cas, pour des raisons de compatibilité avec les couches inférieures on demandera à l'utilisateur :

- a - de se conformer aux primitives de fonctions de base (§ 5.4.1).
- b - de fournir le format correct des messages selon l'action demandée pour que le moniteur puisse les découper en paquets, segments (bornés ou libres) conformément aux exigences du réseau (§ 5.4.2).

Par exemple:





#### 5.4.1 - LES PRIMITIVES DU NIVEAU 7

En fait, au niveau 7, l'utilisateur disposera de quatre primitives de type:

OPEN..(CON) avec les paramètres d'adressage source/destination au niveau du réseau (L3) ou de la session (L4). Il occupera la zone  $Z_E$  et attendra les accusés de réception dans la zone  $Z_L$ .

CLOSE..(DISC) avec les paramètres d'adressage et éventuellement de causes. Il provoquera la libération au niveau de L3 après RAZ de toutes les couches intermédiaires. Il occupera la zone  $Z_E$  s'il vient de l'appelant. S'il vient de l'appelé, il provoquera la même libération, si l'appelant l'autorise. L'appelé en sera averti après libération totale de ses couches intermédiaires (dans la zone  $Z_L$ ).

- SEND
- a - pour les données (avec comme paramètre celui d'adressage du destinataire placé dans la couche  $Z_E$ )  $Z_E$  s'enrichira des autres paramètres de couches durant le transfert à travers celles-ci.
  - b - pour les ACK (ACC) ou ACK (REJ) a un appel entrant donné (avec en principe le seul paramètre de destination à placer dans  $Z_E$ ): Il sera émis tel quel, étant donné que l'octet qui l'enregistre-  
ra transitera directement à travers les couches jusqu'à la couche réseau.

RECEIVE permettra à l'utilisateur informé par l'octet d'information que quelque chose est à prendre dans la zone  $Z_L$ .

Il s'agit soit:

- a - d'un appel entrant.
- b - d'accusés de réception à un appel sortant émis précédemment.
- c - de données pures.

Notons que les commandes ou les réponses de tout ce qui concerne les niveau 4, 5, 6 font uniquement appel aux programmes "Moniteur" et "Interface Réseau" et non au programme d'application.

#### 5.4.2 - FORMATS

##### 1 - Synchrones (X21)

S	S	S	ENTETE	E	C	B
Y	Y	O	(Début)	T	R	C
N	N	H		B	C	C

S	ENTETE	S	TEXTE	E	C	B
O	(suite et	T	(début)	T	R	C
H	fin)	X		B	C	C

S	ENTETE	S	TEXTE	E	C	B
O		T	(fin)	T	R	C
H		X		X	C	C

L'entête comporte:

- adresse source (AD S)
- adresse destination (AD D)
- code type de message, si protocole BSC:
  - APPEL = ENQ/ACK, NAK
  - TEXTE = DATA/ACK, NAK
  - LIBERATION = EOT



## Remarques:

- 1 - La longueur est variable (car délimiteurs: STX, ETB/ETX).
- 2 - Les caractères CRC/BCC sont ajoutés au niveau physique.

2 - Asynchrone

Le format est identique au cas 1, seulement entre chaque caractère on a START /STOP pour l'autosynchronisation.

3 - Paquet X<sub>25</sub>: longueur maximum 128 bytes

Appel

C	A	AD	AD	COMPLEMENTS
O	D			
D				
E	D	S	D	

Libération

C	AD	RAISONS
O		
D		
E	D	

Données

C	AD					
O						
D		P	P	M	Q	DATA
E	D	(R)	(S)			

Acc. REC  
RR ou RNR

C	AD	P
O		
D		
E	D	(R)

#### 4 - Ethernet. Longueur d'un segment

AD	AD	T				C
D	S	y	L		DATA (50 à 1500 B)	R
		p				C
		e				

Remarque: Le type du message peut être .....

- Appel
- Libération
- Donnée
- Statistique

Remarquons que le formatage est fonction de deux choses:

- le type d'opérations (Appel, données, ACK/NAK, libération)
- le réseau qui les acheminera soit:
  - protocole HDLC
  - semi-protocole ETHERNET
  - procédure X21
  - asynchrone X20.

En asynchrone, on pourrait choisir entre

- le plus protégé: HDLC sous la forme paquet X25
- le moins protégé: X21.

A ce sujet, nous proposerons au chapitre VI un alignement des protocoles pour pouvoir interfacer tous les réseaux avec les couches supérieures (4 à 7) via un "BUS" (multibus) ou une "mémoire commune".



## CHAPITRE VI - INTERFACE BANALISE

=====

### 6.1 - COMPATIBILITÉ ENTRE RÉSEAUX

#### 6.1.1 - Interfaçage entre réseaux

##### 6.1.1.a - au niveau du terminal

##### 6.1.1.b - au niveau du réseau

### 6.2 - PROTOCOLE DE BASE ALIGNÉ SUR X.25 ET LES PRIMITIVES D'INTERCOUCHES

#### 6.2.1 - Introduction

#### 6.2.2 - Sélection des primitives X.25

### 6.3 - CONVERSION DES PRIMITIVES X.25 ET OCTETS D'ACTIONS (ET INVERSEMENT)

#### 6.3.1 - Principe

#### 6.3.2 - L'octet d'information (Rappel)

#### 6.3.3 - L'interface banalisé

##### 6.3.3.1 - objectifs

##### 6.3.3.2 - Principe d'interfaçage

### 6.4 - ALIGNEMENT DE X.21 SUR X.25

### 6.5 - ALIGNEMENT D'ETHERNET SUR X.25

### 6.6 - ALIGNEMENT ASYNCHRONE SUR X.25

### 6.7 - ALIGNEMENT DES PROTOCOLES

## C H A P I T R E    V I

---

### I N T E R F A C E    B A N A L I S É

#### 6.1 - COMPATIBILITÉ ENTRE RÉSEAUX

##### 6.1.1 - INTERFACAGE ENTRE RESEAUX

Il est évidemment indispensable de trouver une solution au problème de la compatibilité des réseaux afin de permettre à tous les terminaux de se comprendre, quel que soit le réseau dans lequel ils se trouvent.

Deux solutions sont admises par les instances internationales:

a - Au niveau du terminal

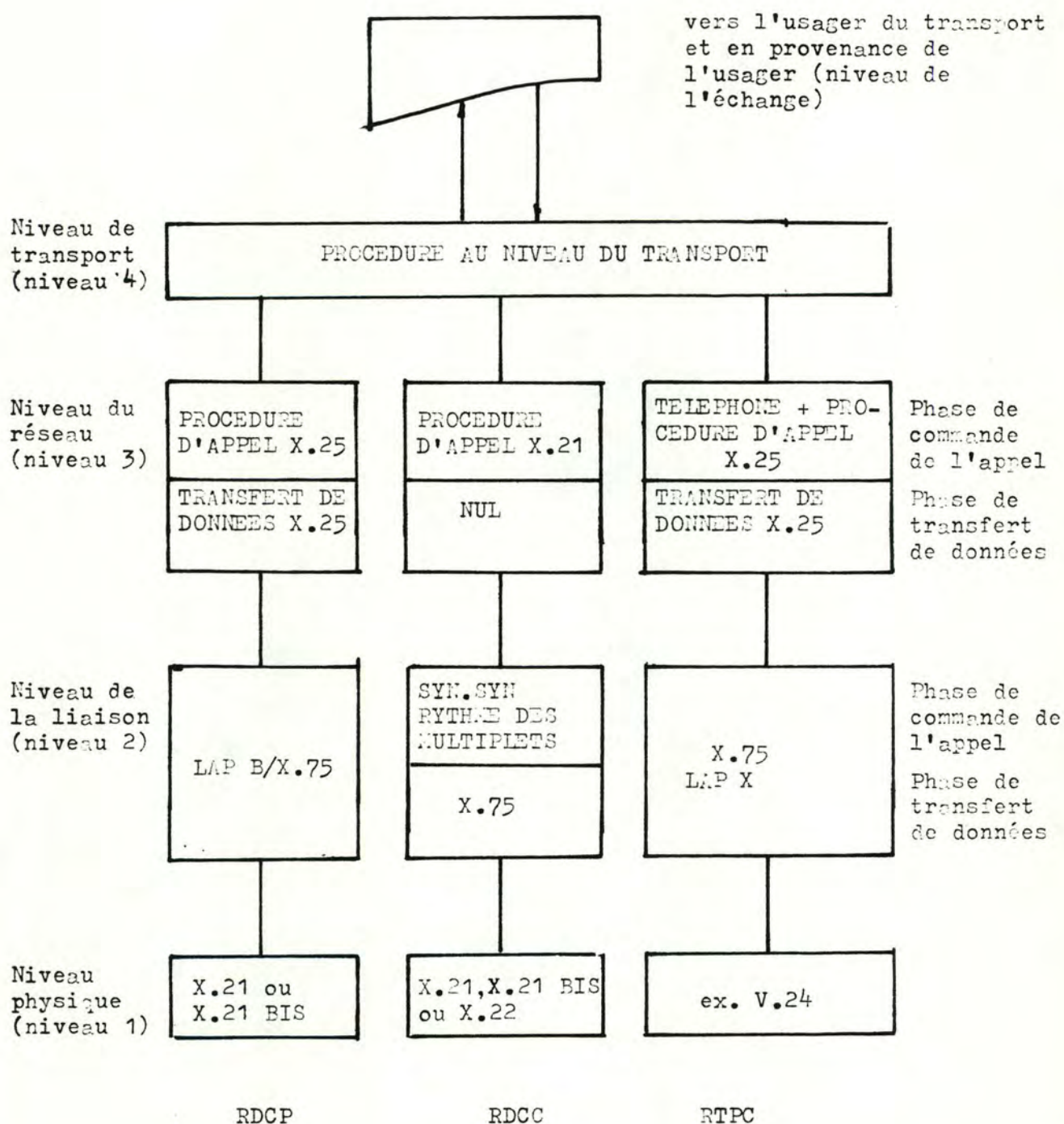
+ L'introduction de couches d'adaptation pour assurer la banalisation via une couche commune de transport comme proposé à la figure 6.1.1.

+ La solution proposée par le CCITT voudrait amener, autant que faire se peut, les terminaux des réseaux

- de commutation de circuits à procédure synchrone (X21) ou asynchrone (X20)
- de commutation téléphonique à procédure physique élémentaire
- de types locaux, qu'ils soient annulaires, à segments ou à jetons...

à se "comprendre" grâce à la version X25 de l'HDLC.





## INTERFACE RESEAUX

- Fig. 6.1.1 -

+ La proposition de la CCITT est illustrée à la figure 6.1.1.

+ Comme :

- d'une part, l'interface est complètement transparente et de type "store and forward" et sa fonction essentielle est de réaliser la transformation des formats, eux-mêmes, parfaitement normalisés,

- d'autre part, les deux premiers types de commutation se font sans protocole de recouvrement d'erreurs, tandis que ceux des réseaux locaux disposent d'un protocole ou non suivant les exigences du concepteur,

l'alignement de protocoles ou procédures faibles sur X25 reste imparfait, car les deux fonctions essentielles sont reportées au niveau 4 de transport.

Ces deux fonctions sont :

- l'établissement et la libération des liaisons aux niveaux supérieurs,
- la récupération des erreurs de données impossible aux niveaux 3 de X25 et X21.

Malgré les remarques et les spécifications complémentaires, cette normalisation est difficilement implémentable. En effet, l'interfaçage entre les différentes couches L3



et le niveau de transport L4 ne peut être envisagé de la même façon pour les divers cas. C'est la raison pour laquelle en adoptant cette recommandation (CCITT) nous avons introduit entre L3 et L4 une couche "réseau" (aujourd'hui en voie d'adoption par les instances internationales). Cette nouvelle couche pourra également intervenir pour opérer des relations de type "gateway".

Le schéma de l'interconnexion entre terminaux de réseaux différents est illustré à la figure 6.1.2.

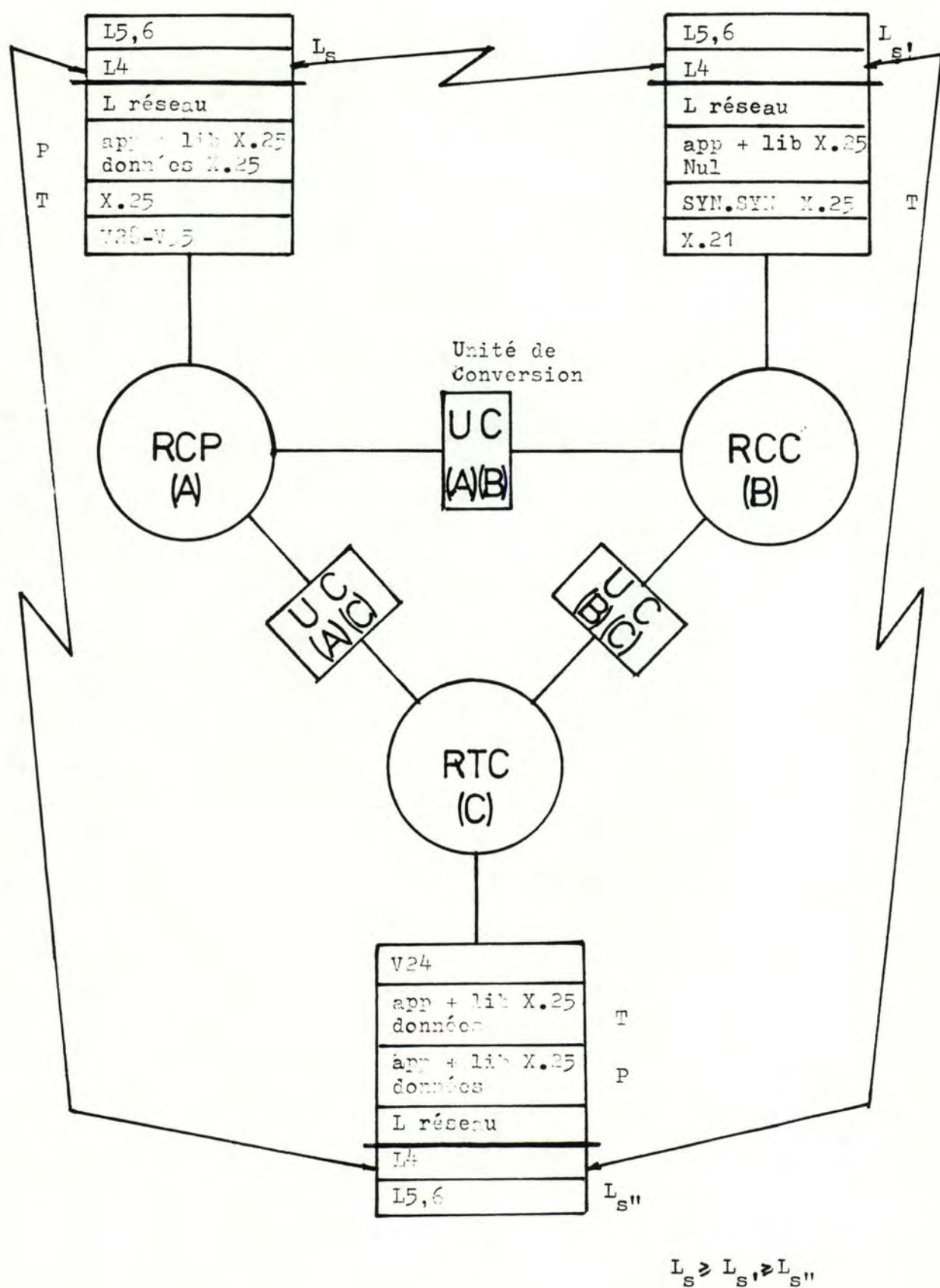
#### b - Au niveau du réseau

+ La liaison s'opère via une unité de conversion interfaçant 2 à 2 les réseaux à leur frontière commune comme proposé à la figure 6.1.3.

+ Cette solution prend pour base l'alignement sur le réseau le plus pauvre des autres réseaux, ce qui peut paraître illogique et irrationnel. La raison est que la banalisation ou la normalisation au niveau international pourra mieux se faire sans ambiguïté.

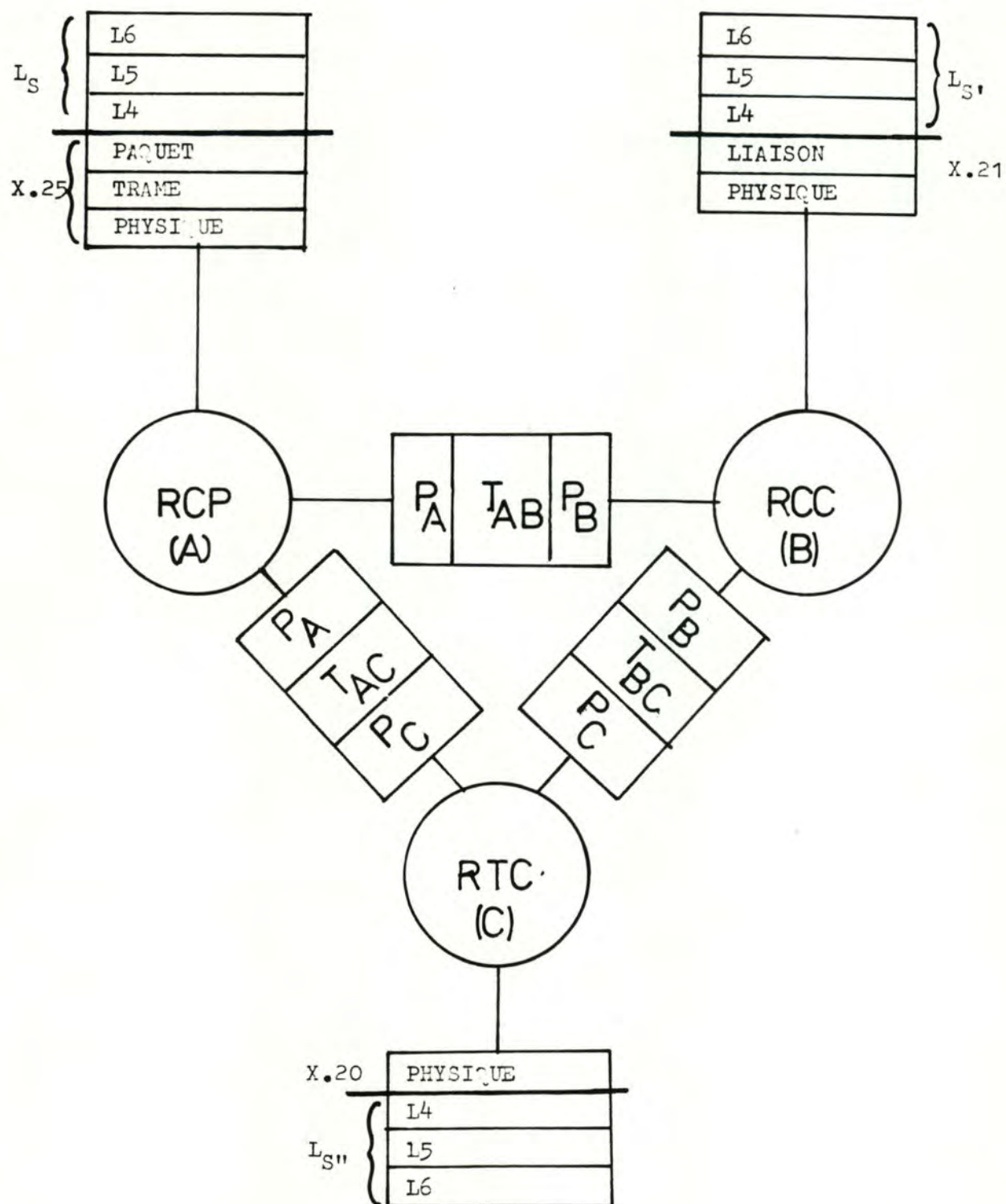
+ Cette interfaçage entre réseaux illustré à la figure 6.1.3 comporte les caractéristiques suivantes:

- Il peut être de type "store and forward", via uniquement la mémoire centrale ou non. Dans ce dernier cas, des unités de disques doivent être introduites, leur importance dépendant du trafic.



- Fig 6.1.2 -





$$L_S = L_{S'} = L_{S''}$$

- Fig. 6.1.3 -

- La première version, moins dispendieuse, peut être envisagée lorsque le transcodage et le reformatage peuvent être réalisés "à la volée" quel que soit le trafic.
- Le problème du changement de débit lors du passage d'un réseau à l'autre exige que les vitesses de transmission des réseaux autres que le RCP soient inférieures à la vitesse de ce dernier. Dans ce cas, le contrôle de flux peut se faire par un RR/RNR au niveau 3 du RCP.

+ L'interface comporte 3 couches ( $P_A$ , T,  $P_B$ ). Examinons leurs fonctions:  $P_A$  et  $P_B$  sont destinés à réaliser l'écho des commandes et réponses des protocoles respectifs, tandis que T assurera la transparence ou la conversion selon le cas et dans les deux sens.

+ Les couches 4, 5 et 6 sont en principe identiques dans les réseaux face à face, les couches "réseau" gardent leur spécificité.

+ A titre d'exemple, la solution d'interfaçage RCP - R.Telex illustrée aux figures 6.1.4 a et b est celle qui permet d'assurer des échanges entre terminaux Teletex (sur RCP) et Telex (sur R-Telex).

+ Le réseau telex, en effet, est un réseau qui:

- ne dispose pas de protocole. Un simulateur d'écho dans l'interface devra générer ce qui est nécessaire au dialogue avec le RCP.



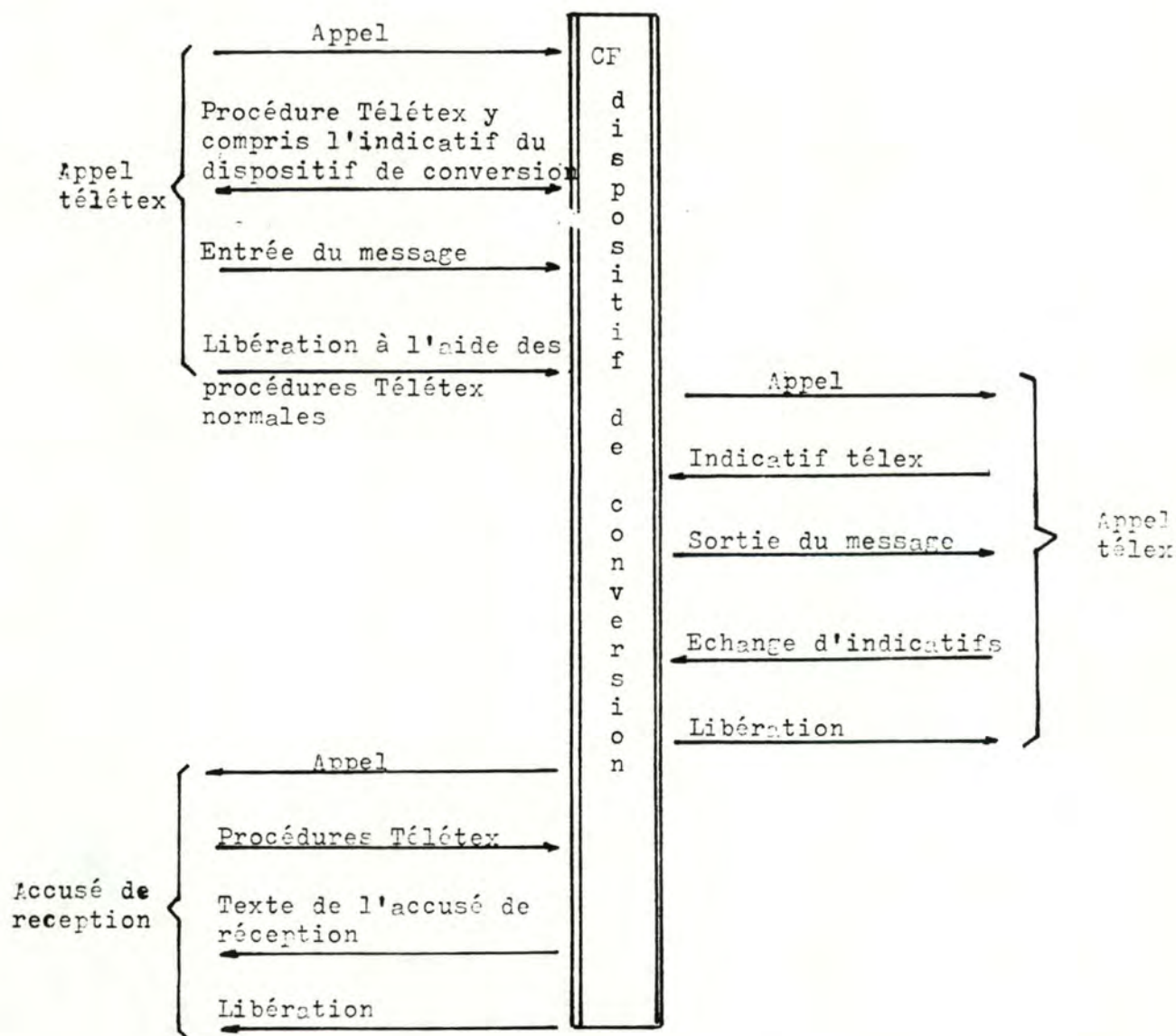
- dispose d'une riche procédure qui devra être soit simulée, soit convertie en X25 par l'interface pour les deux sens de transmission.

+ Cette unité de conversion entre deux récepteurs RCP et R-télex doit assurer des relations entre des terminaux (Teletex et Teletypes publics) apparentés par la possibilité qu'ils ont de transmettre des textes.

+ C'est un cas extrême d'interfaçage entre un réseau à haut niveau de performance (RCP - X25) et un réseau complètement transparent (R.telex) qui d'ailleurs ne pourra accepter pour ses terminaux les couches 4, 5, 6.

RESEAU TELETEX

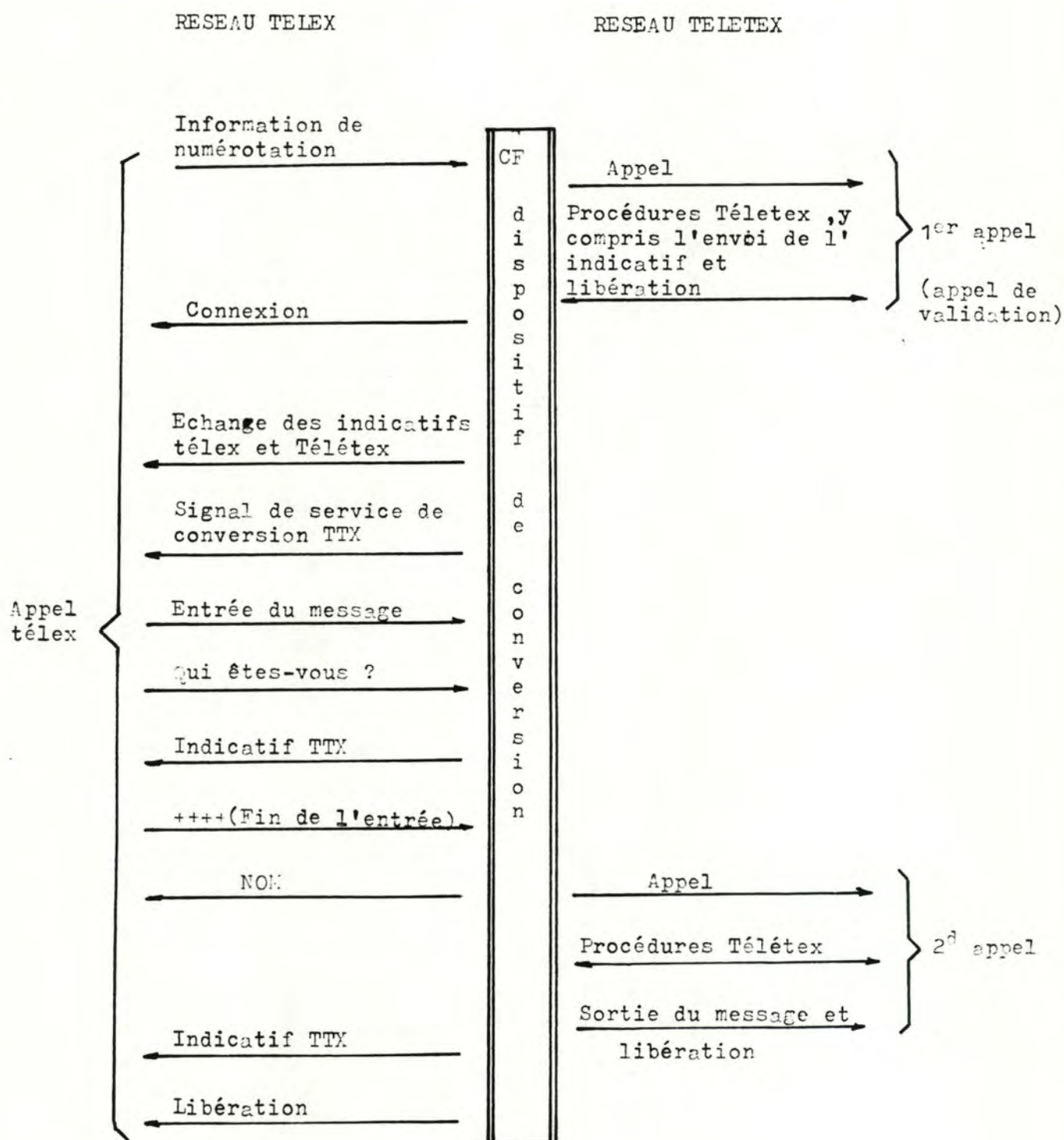
RESEAU TELEX



UNITE DE CONVERSION

- Fig. 6.1.4<sub>a</sub> -





UNITE DE CONVERSION

## 6.2 - PROTOCOLE DE BASE ALIGNÉ SUR X.25

### ET LES PRIMITIVES D'INTERCOUCHES

#### 6.2.1 - INTRODUCTION

La recherche de la compatibilité des réseaux devra donc reposer sur le principe que les réseaux pauvres du point de vue protocoles et procédures doivent s'aligner sur le plus nanti que nous ayons choisi, c'est-à-dire X.25.

Ceci implique la nécessité d'une part, de supprimer des primitives non fondamentales en X25 (cette suppression, notamment des primitives de la réinitialisation pourra être réexaminée dans la suite), et d'autre part, d'enrichir ou de traduire les commandes des réseaux les moins dotés.

Il y a donc lieu de réexaminer dans le détail les primitives de protocoles et de procédures du RCP en se rappelant que nous nous trouvons dans un contexte de réseau local et non de réseau public. Ensuite, nous proposerons l'assimilation des primitives sélectionnées au cas des réseaux les plus représentatifs, tels que ceux :

- en mode synchrone (X 21)
- en mode asynchrone (X 20)
- mixte (Ethernet).

On trouvera dans ce qui suit, sous des formes souvent redondantes mais parfois complémentaires, les primitives d'un réseau X25 et la façon dont elles peuvent être associées aux primitives d'intercouches.



Ces primitives ne sont pas toutes utiles lorsqu'on n'évolue pas dans le contexte d'un réseau de commutation par paquets. Les différents schémas ont pour but de présenter celles qui sont fondamentales et par réciprocité, celles qui sont "accessoires".

### 6.2.2 - SELECTION DES PRIMITIVES X25

La figure 6.2.1 illustre la topologie des séquences d'un réseau RCP de commutation par paquets.

On découvre les séquences et les primitives:

- fondamentales:

A = appel avec les accusés de réception  
(ACC et REJ ou LIB)

B = libération

C = données

- complémentaires

D = réinitialisation

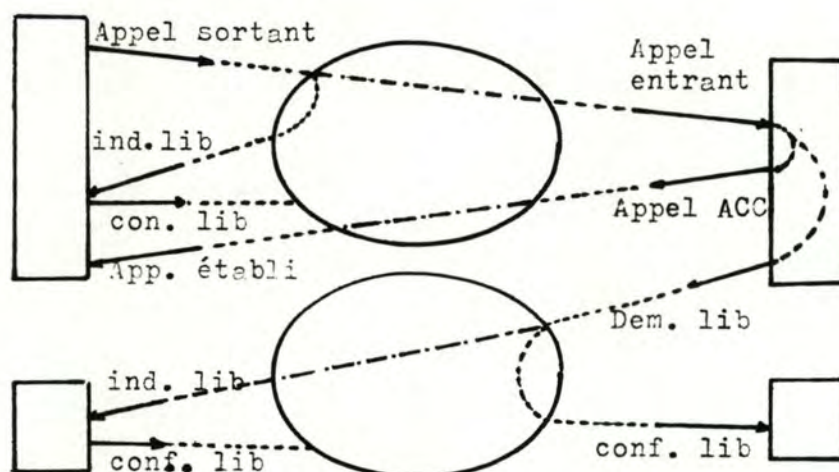
E = reprise.

Pour les trois types de raisons qui suivent, un certain nombre de primitives de ces séquences seront ignorées.

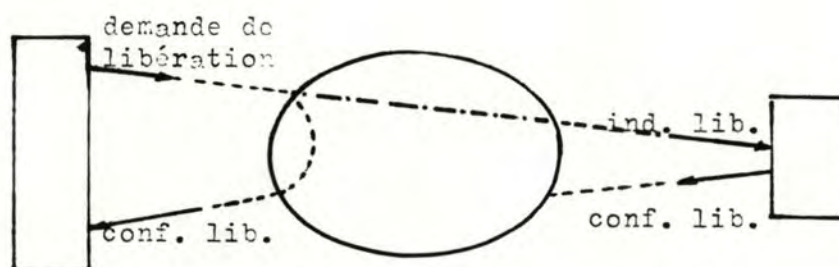
#### 1ère raison:

le fait que, pour notre application, X25 constitue un protocole d'échanges de point à point ou plus exactement de stations à stations. X25 n'est donc pas exploité dans un réseau à noeuds de commutation. Ceci justifie la sélection des primitives encadrées dans la figure 6.2.2 pour les séquences de base. Ces primitives sont indispensables, les autres sont inhérentes à l'existence du réseau RCP.

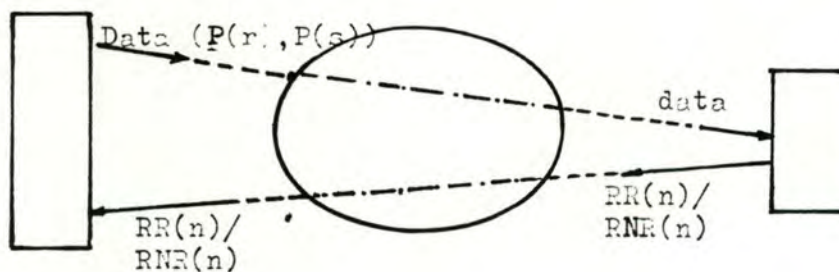
## A. Appel



## B. Libération



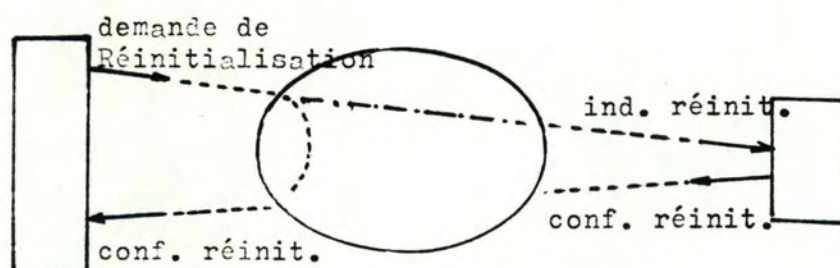
## C. Données



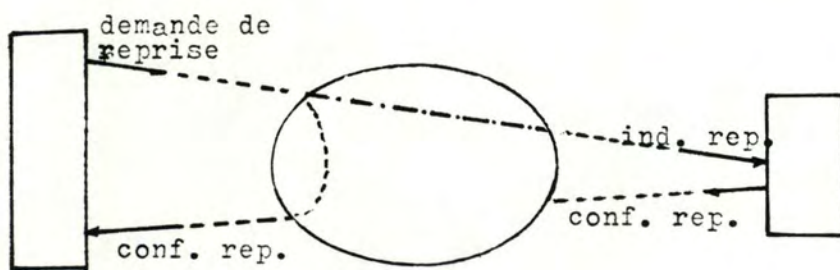
NIVEAU PAQUET ----- 13



## D. Réinitialisation



## E. Reprise

NIVEAU PAQUET  $\longleftrightarrow$  L3- Fig. 6.2.1<sub>b</sub> -

## A. GESTION DES COMMUNICATIONS.

## ETABLISSEMENT

<u>APPEL</u>							
0	0	0	1				
0	0	0	0	1	0	1	1
LAD 1		LAD 2					
ADRESSE							
				0	0	0	0
0	0	0	0	0	0	0	0
DONNEES							
(lg max 16 oct)							

<u>CONFIRMATION</u>							
0	0	0	1				
0	0	0	0	1	1	1	1

## LIBERATION

<u>LIBERATION</u>							
0	0	0	1				
0	0	0	1	0	0	1	1
CAUSE							

## B. TRANSFERT DE DONNEES ET CONTROLE DE FLUX.

<u>DONNEES</u>							
Q	0	0	1				
P(R)		M	P(S)		O		
DONNEES							

Q: quantificateur  
de données.  
M: données à  
suivre

PRET A RECEVOIR (RR)

0	0	0	1				
P(R)		0 0 0 0 1					

NON PRET A RECEVOIR (RNR)

0	0	0	1				
P(R)		0 0 1 0 1					

INTERRUPTION

0	0	0	1				
0	0	1	0	0	0	1	1
DONNEES							

REINITIALISATION

0	0	0	1				
0	0	0	1	1	0	1	1
CAUSE							
DIAGNOSTIC							

CONFIRMATION

0	0	0	1				
0	0	1	0	0	1	1	1

CONFIRMATION

0	0	0	1				
0	0	0	1	1	1	1	1

## C. REPRISE.

REPRISE

0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	1
CAUSE							

CONFIRMATION

0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1



2ème raison:

Les primitives RR (ACK+) et RNR (ACK-) de la séquence de données (C) sont utilisées par X25 au niveau 3 pour le contrôle de validité et de séquençement des paquets. Elles peuvent être ignorées au niveau 4, sauf en cas de blocage du protocole. Cela provoquera soit un relâchement total, soit la ré-initialisation ou la reprise au niveau 3 qui devrait être synchronisée au niveau 4.

Toutefois, les réseaux X21, Ethernet, ... démunis de ces primitives de X25 devront adopter un protocole sous-jacent dans la couche 3, avec le même mécanisme des ACK<sup>+</sup> pour conserver le même parallélisme de fonctionnement pour tous les supports de données.

3ème raison:

Les primitives de réinitialisation et de reprise des séquences (D) et (E) pourraient être exploitées au niveau supérieur de X25 (cfr Ch. VII). Elles ne le seront pas, car elles devraient être introduites dans X21, Ethernet... sous la forme d'un protocole permettant de réaliser le "Restart" et le "Reset" comme pour X25, ce qui serait disproportionné par rapport à l'ensemble de leurs logiciels.

Enfin, toutes ces considérations nous conduisent à la sélection des primitives essentielles de X25 répertoriées dans la figure 6.2.1 pour les séquences (A), (B) et (C) où les séquences (D) et (E) ont été rebutées.

Rappelons également que la figure 6.2.2 reprend le format, le code et le contenu des différents paquets de X25.

On trouvera de façon détaillée à l'annexe G:

- la signification
- le format
- les paramètres

des différents paquets de X25.



## 6.3 - CONVERSION DES PRIMITIVES X25 EN OCTETS D' ACTIONS (ET INVERSEMENT)

### 6.3.1 - PRINCIPE

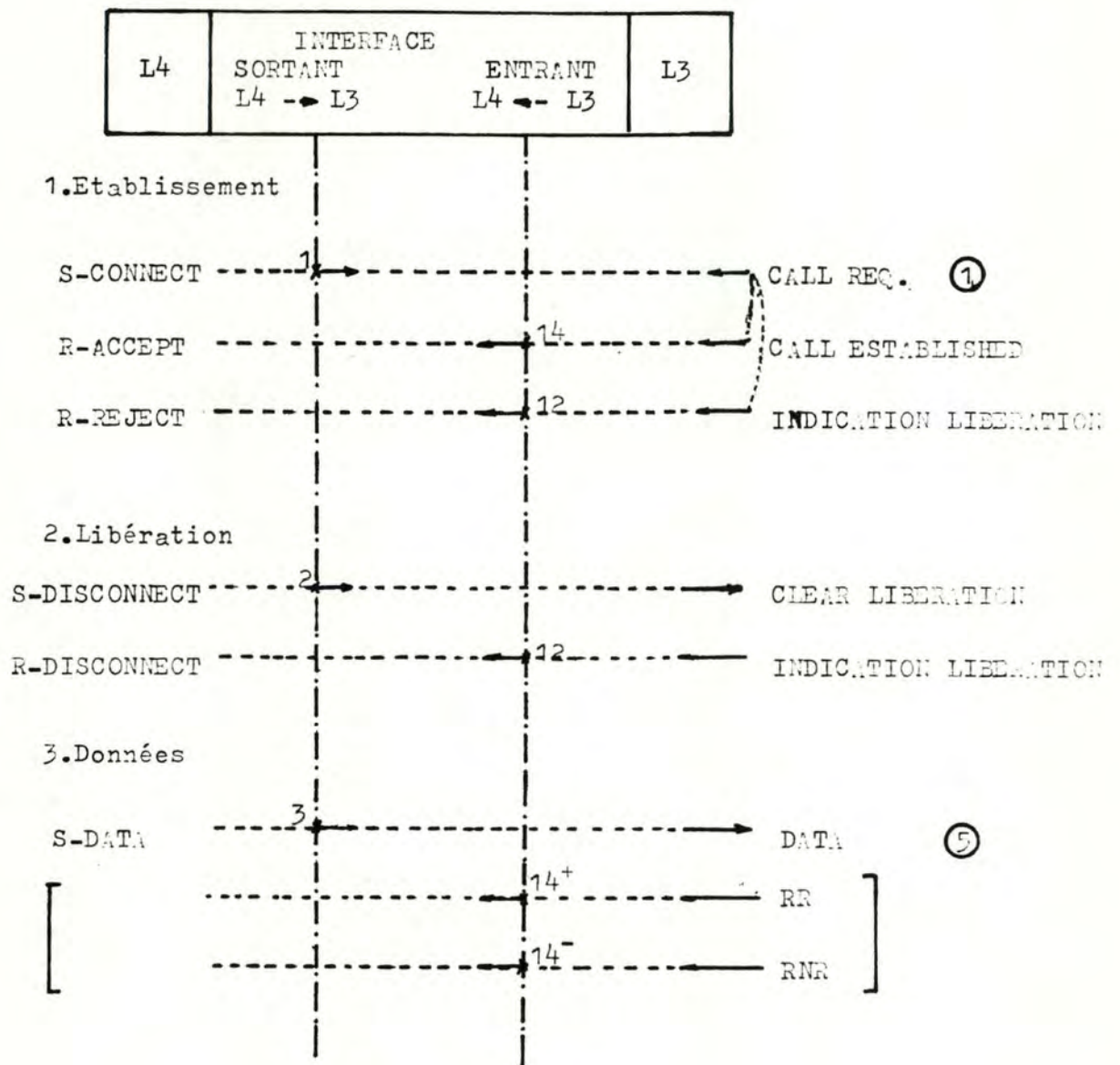
Dans le paragraphe précédent, nous avons fait la sélection des primitives fondamentales de X.25 se situant au niveau L3. Elles émanent des couches supérieures via l'interface réseau (IR) ou proviennent de L3 pour transiter vers les couches supérieures via IR.

L'intercouche aura pour rôle de mémoriser l'action de ces primitives par un des bits de l'octet d'actions ou d'informations entrant (E) ou sortant (S).

Aux figures 6.3.1 et 2, on a indiqué respectivement pour l'appelant et pour l'appelé, les relations entre les primitives X25 (chiffres cerclés) et les primitives de couche (références non cerclées) ainsi que les positions de bits correspondants dans les octets entrant (E) ou sortant (S).

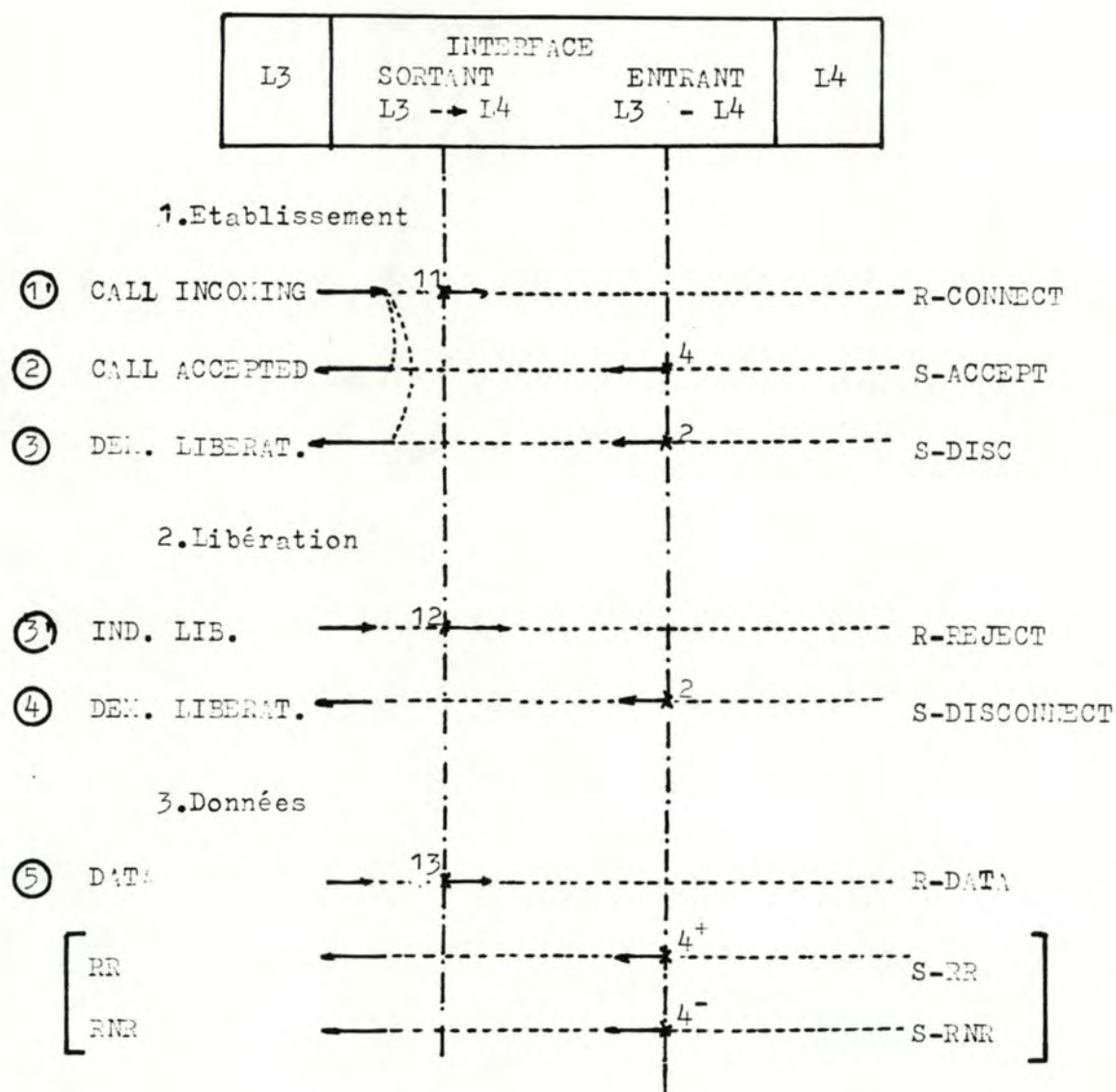
On trouve regroupés en deux octets la position et le sens des primitives propres à la figure 6.3.3.

Enfin, à la figure 6.3.4, nous trouvons l'ensemble des primitives de couches groupées par famille et non par sens. Elles seront paramétrées pour remplir complètement leur rôle dans les différentes couches.

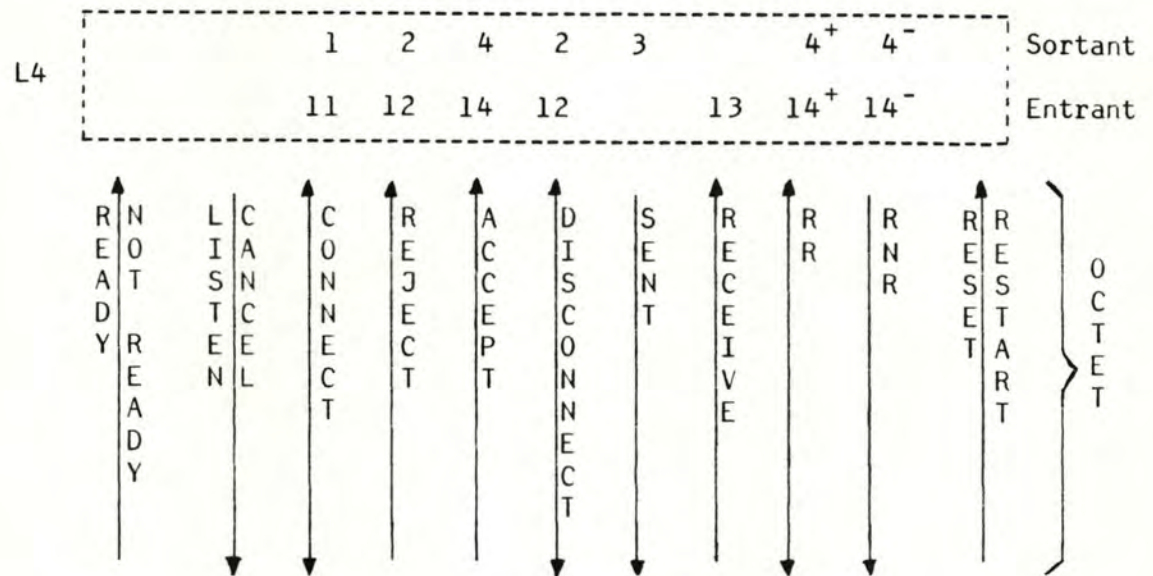


- Fig 6.3.1 INTERFACE L3/L4 : COTE APPELANT -





- Fig. 6.3.2 INTERFACE L3/L4 : COTE APPELE -



Parmi toutes ces primitives, celles qui concernent l'initialisation du terminal ainsi que la synchronisation (reprise ou réinitialisation), ne sont pas reprises dans la suite.

Voici celles qui ont été retenues:

SENS L4	L3	CONNECT
		DISCONNECT
		SEND DATE
		RR / RNR
SENS L3	L4	CONNECT
		REJECT
		ACCEPT
		DISCONNECT
		RECEIVE DATA

- Fig. 6.3.3. -



PAQUET (L3)                      TRANSPORT (L4)

-----

Transfert d'information pour indiquer ou remplir le mot d'état (octet). Il existe quatre classes de primitives d'intercouches.

1 - Etablissement et terminaison

- CONNECT                      (établir l'association)
- LISTEN                      (prêt à recevoir un connect)
- ACCEPT                      (association acceptée)
- REJECT                      (association rejetée)
- CANCEL                      (neutralise listen)
- DISCONNECT                  (ferme l'association)

2 - Transfert de données

- SENT                      (RTS demandant)
- RECEIVE                      (RTR montant)
- PR                      (prêt à recevoir)
- RNP                      (non prêt à recevoir)

3 - Synchronisation

- RESET                      (réinitialisation)
- RESTART                      (reprise)

4 - Initialisation du terminal

- READY
- NOT READY

### 6.3.2 - L'OCTET D'INFORMATIONS

L'ensemble des opérations du réseau va être orchestré par un "moniteur de gestion et (éventuellement) d'interruptions". Celui-ci va gérer, entre autres, le passage de L3 (quel que soit le protocole ou la procédure utilisé: X21, Ethernet...)

dans la couche "NETWORK LAYER" (voir paragraphe suivant).

Pour cela, deux "octets" vont être créés:

- un pour le sens L4 → Network layer (IR)
- l'autre pour le sens L3 → Network layer (IR)

Lorsqu'une primitive arrive, de L4 ou de L3 cela va provoquer la mise à 1 d'un bit particulier dans l'octet correspondant. Le bit correspond à l'action demandée par la primitive. L'opération inverse, (c'est-à-dire à partir d'un bit, générer la primitive correspondante), aura lieu pour le passage de L3 → L4.

Remarquons, dès à présent, que l'octet d'information ne permet qu'une interprétation générale de la donnée (bit 0 ou 1). Ces octets ne communiquent d'un niveau à l'autre que la présence ou non d'une opération réalisée ou à réaliser. La liste des paramètres, et des primitives se trouve dans une zone réservée. Les paramètres ou les informations associés à cette action seront transférés ensuite par le "moniteur".

Il existe 4 zones réservées, mises à des adresses symboliques fixes:

- BOU:     byte d'intercouche sortant du niveau 7 se transformant en BONR au niveau IR.



- BIU: Byte d'intercouche entrant au niveau 7. Il provient d'une transformation de BINR au niveau IR.
- ZE: Zone d'écriture de la couche supérieure.
- ZL: Zone de lecture de la couche supérieure.

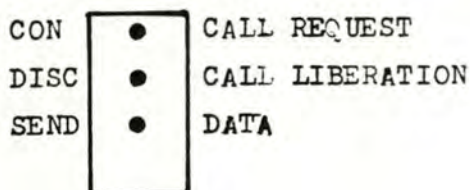
Chaque primitive va provoquer une "interruption" suivie d'un déroutement vers le "moniteur" après sauvetage. Une scrutation peut remplacer l'interruption si le trafic le permet. On effectue alors le traitement correspondant, puis on restaure le contexte.

Il s'agit donc de remplir une zone ZE (zone d'écriture) de primitives avec paramètres. Lorsque celle-ci est remplie, il y a positionnement d'un bit qui est lu périodiquement, provoquant une interruption pour l'envoi du segment contenu dans ZE.

En d'autres mots, cette zone reçoit des informations de différents niveaux du programme. Le moniteur, dont elle constitue une partie, en informe le niveau réseau L3 par l'intermédiaire de l'octet d'interface, schématisé ci-dessous.

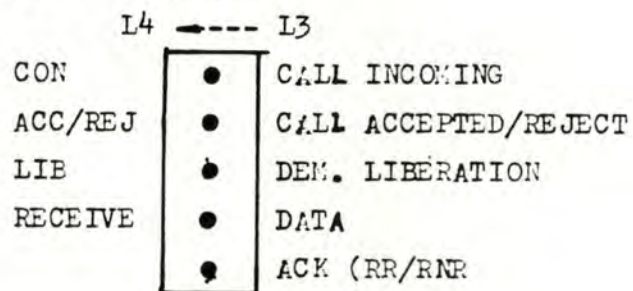
#### OCTET EN SORTIE

L4 ----- L3



De la même façon, un octet et une zone sont disponibles en entrée (sens L3 → Network layer). La zone est alors une zone de lecture ZL, installée à une adresse fixe et connue. Celle-ci reçoit du réseau des paquets validés. Une interruption va alors être créée pour informer le moniteur de l'arrivée d'un paquet. Ce dernier connaît ainsi le moment où un segment complet est reçu. Une file d'attente de segments peut être, par ailleurs, constituée en réception. Le passage de l'information s'effectue grâce à un octet schématisé ci-dessous.

#### OCTET EN ENTREE



### 6.3.3 - INTERFACE BANALISE

#### 6.3.3.1 - Objectifs

Pour pouvoir connecter des équipements de différents "constructeurs", l'ISO a défini une norme d'architecture de réseaux informatiques où peuvent prendre place les protocoles qui permettront la constitution de réseaux hétérogènes. Cette architecture connue sous le nom de "modèle de référence pour l'interconnexion de systèmes ouverts" est constituée des sept couches suivantes: application (7), présentation (6), session (5), transport (4) pour les couches supérieures et réseau (3), liaison de données (2), physique pour les niveaux inférieurs.



L'objectif est de permettre la compatibilité des couches supérieures (L7 à L4) avec les couches inférieures assurant la liaison au réseau, quel que soit la nature de son protocole ou de sa procédure.

#### 6.3.3.2 - Principe d'interfaçage

L'émulation ou la banalisation se fera par un alignement optimal avec le protocole le plus élevé (X25) des autres procédures: synchrone (X21), asynchrone et Ethernet... On essaie de présenter L4 en X21, asynchrone, Ethernet comme il se présenterait en RCP.

Un projet de solution a été développé par le CCITT, pour les passages et les transferts entre réseaux de types différents. Pour fixer les idées, reprenons ceci dans la figure 6.3.3.1. Le problème posé est celui d'un terminal ou d'une station se situant sur un réseau de commutation:

- soit RCP
- soit X21
- soit Ethernet
- soit asynchrone.

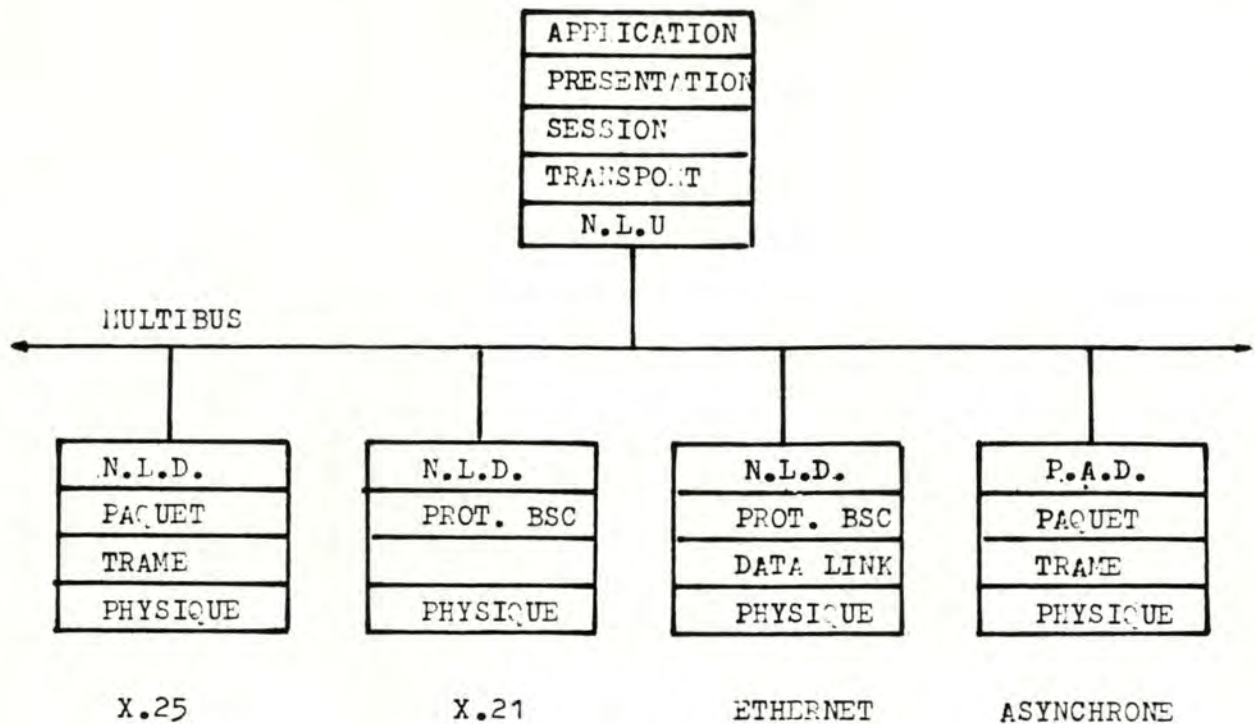


Fig 6.3.3.1

On va donc, pour banaliser les échanges, introduire une "couche interface" supplémentaire (couche hachurée sur la figure 6.3.3.1) que nous appellerons "NETWORK LAYER" (NL) qui, comme on le verra, se divisera en deux sous-couches NLU (up) et NLD (down) séparées par un "multibus".

Les fonctions que doit assurer le "NETWORK LAYER" sont:

- le transfert de l'octet de signalisation par des messages en entrée et en sortie de et vers L4, la couche de transport.
- interpréter les composants des différents formats et les adapter:
  - en format standard pour transférer les informations vers les couches supérieures.
  - en format propre à la procédure destiné à être transféré en sortie vers le réseau.



Rappelons que le problème posé n'est pas de transformer un format d'informations en un autre d'une autre procédure, mais de les convertir tous en une forme "standard" pour être interprétée de façon identique par les niveaux 4 et supérieurs de la station.

L'alignement des différents protocoles et procédure, à savoir X21, Asynchrone et Ethernet sera envisagé dans les paragraphes suivants de ce chapitre.

## 6.4 - ALIGNEMENT DE X21 SUR X25

La procédure X21 comporte deux types de signalisation à savoir:

- une signalisation physique par niveaux de potentiel.
- une signalisation par messages référencés par des primitives paramétrisées, complétant le premier type de signalisation.

Cette signalisation, supportée par 4 fils (figure 6.4.1), selon les potentiels appliqués fait progresser les phases de la séquence et définit la procédure normalisée X21, résumée par des diagrammes que l'on peut trouver en annexe H.

Si le réseau constitué doit s'identifier à un réseau de commutation de circuits à procédure X21, il est indispensable de synchroniser les messages (adressage, identification de la ligne, données...) par des signaux physiques. Il est alors nécessaire de constituer, pour répondre à ces signaux, deux interfaces symétriques au niveau des terminaux appelants et appelés, ainsi que deux interfaces de réseau également symétriques, pour donner l'écho aux signaux provoqués et/ou envoyés par les terminaux.

Cependant, il est courant que les relations entre terminaux soient rendues INDEPENDANTES du réseau physique, c'est-à-dire sans tenir compte des contraintes de la signalisation électrique imposée par la procédure. Il en est d'ailleurs ainsi des protocoles (BSC ou HDLC) où les séquences d'échanges sont uniquement régis par des messages (caractères ou paquets).

C'est la raison pour laquelle une solution possible pour assurer la synchronisation du dialogue serait l'utilisation:



a - de messages, afin de:

- considérer la procédure par messages uniquement pour

- \* l'appel: sortant comme entrant  
(message de sélection),

- \* l'acceptation de l'appel  
(message d'identification de l'appelé)

- d'assurer directement la transmission des données.

b - de la signalisation.

Pour ce qui concerne la demande de libération (fin de communication, appel refusé), celle-ci se fait à l'aide d'une transmission directe appelant-appelé via modem détectant le changement de niveau correspondant. Toutefois, la signalisation électrique étant peu applicable dans les relations inter-stations d'un réseau local, c'est un message qui déterminera la libération de la liaison et de la négociation.

Notons que X21 est caractérisé par le fait qu'il n'y a pas de négociation de la capacité de mémorisation établie entre deux usagers. Si le destinataire ne peut recevoir les informations qu'on lui envoie, ces dernières sont perdues.

De plus il n'y a pas de contrôle de flux possible sur l'information.

De surcroît, pour tout ce qui est d'une part, de la "réinitialisation" ou de la "reprise" spécifique à X25 et d'autre part, l'"indication ou la confirmation de libération" spécifique au réseau, rien n'y est envisagé. Il serait possible de les reporter au niveau des segments de L4 moyennant l'introduction d'un protocole adéquat.

Il en est de même pour le contrôle du séquençement. Si celui-ci ne se trouve pas dans la procédure, on peut le reporter au niveau des segments de L4, en introduisant pour chacun d'eux la numérotation  $N(R)/N(S)$  et en choisissant un modulo 2 exposant  $n$ , correspondant au cadre choisi de la fenêtre (fonction de l'opération et de la vitesse de transmission).

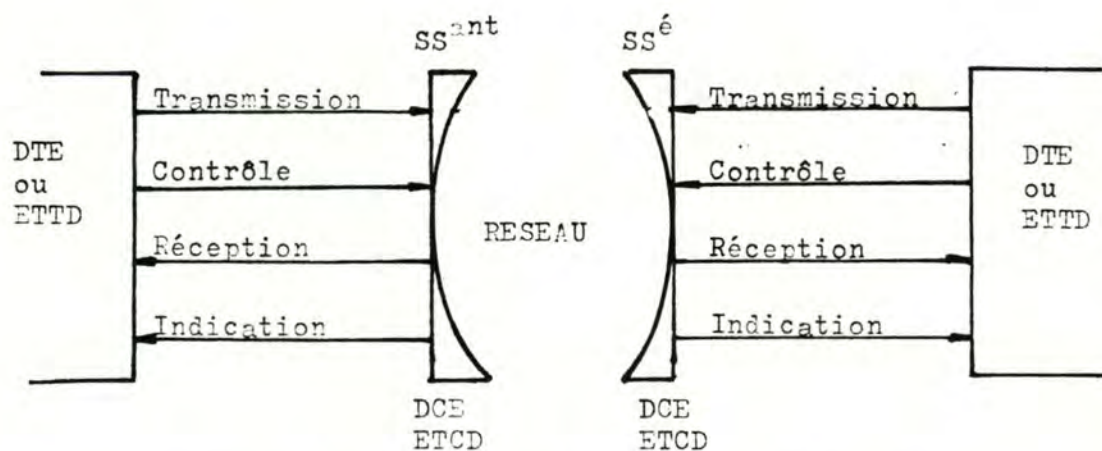
On trouve dans l'annexe H, l'énumération de chacune des étapes de l'échange suivant X21, tel qu'il est décrit dans les diagrammes qui y correspondent.

L'annexe H donne la correspondance entre les onze "types" de paquets repris dans X25 et les phases contenues dans X21.

Les phases fondamentales que nous avons retenues dans X25, devront se retrouver dans X21, tant au départ qu'à l'arrivée (fig. 6.4.2).

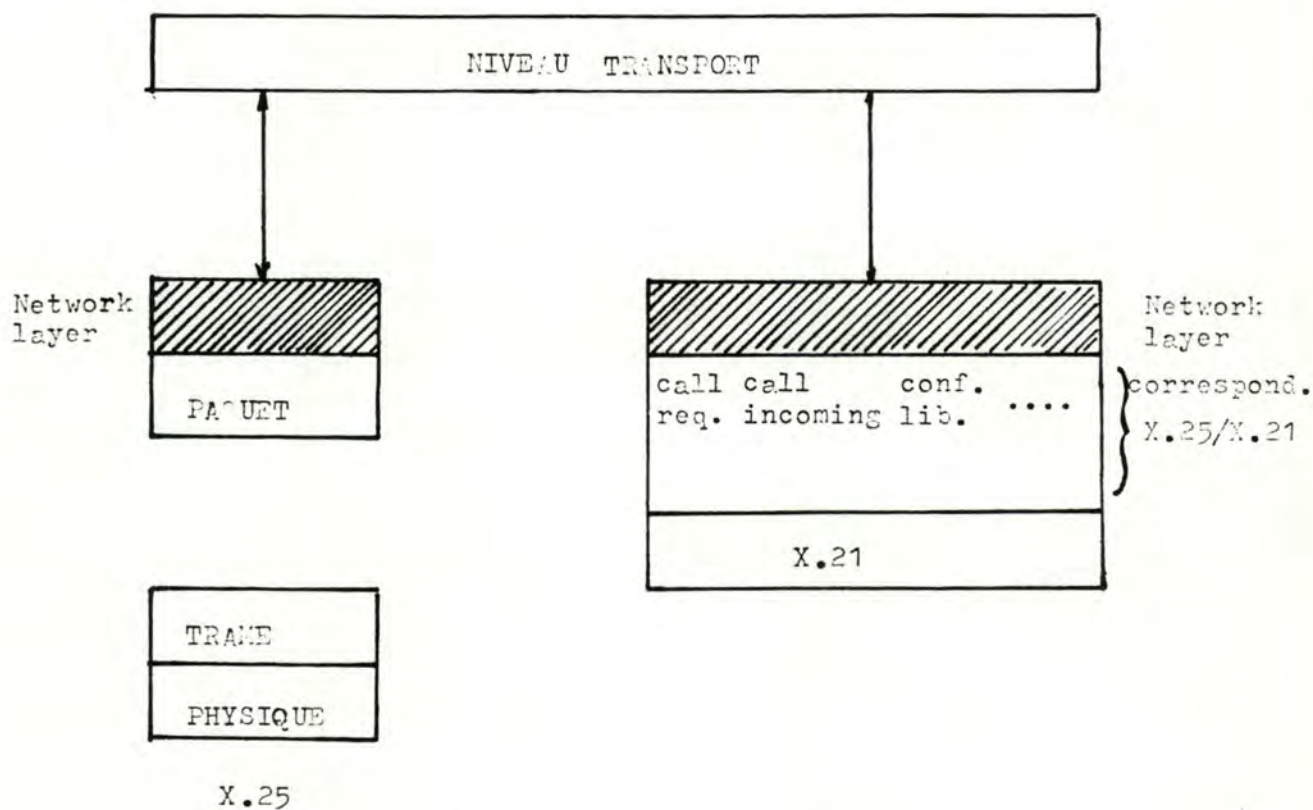
Lors de l'exposé (au paragraphe 6.2.2) de la seconde raison d'alignement, nous avons signalé la nécessité d'introduire au niveau 3 de X21 un protocole pour le contrôle de validité et de séquençement des données. Nous reparlerons de l'introduction de ce protocole au paragraphe 6.7 (Alignement des protocoles).





ETTD = équipement terminal de transmission de données  
 ETCD = équipement de terminaison de circuits de données  
 DTE = data terminal équipement  
 DCE = data circuit terminal équipement

- Fig. 6.4.1 -



- Fig 6.4.2 PRINCIPE DE L'ADAPTATION D'UN TERMINAL X.21 -

## 6.5 - ALIGNEMENT D'ETHERNET SUR X25

Nous n'entrons pas dans les détails du fonctionnement du réseau Ethernet car ce qui nous intéresse présentement c'est l'accès de l'utilisateur au réseau.

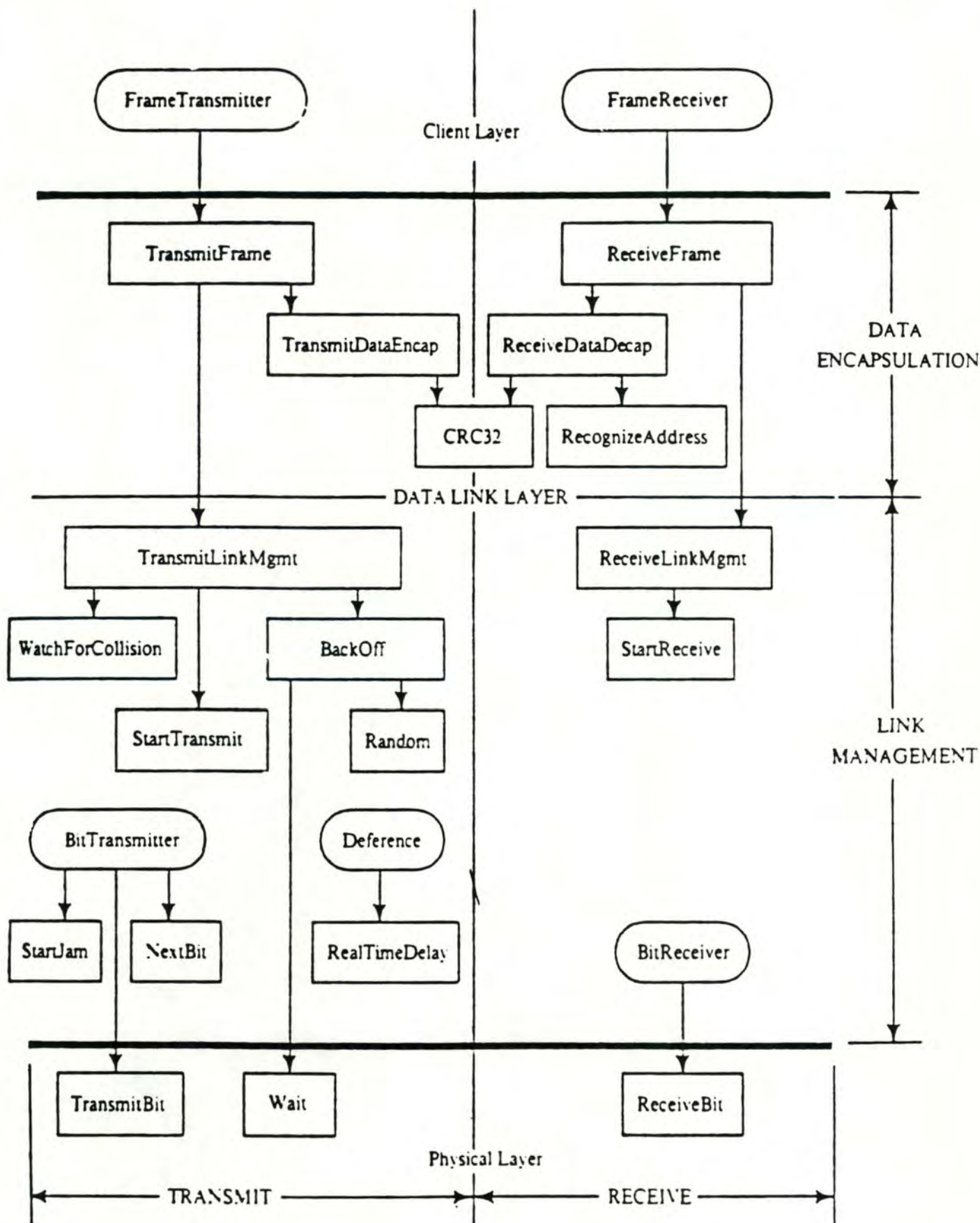
Certaines firmes comme INTEL, DEC, XEROX, ... ont apporté une aide non négligeable à l'utilisateur, sous la forme de circuits intégrés qui remplissent les fonctions dévolues aux deux premières couches du réseau (voir fig. 6.5.1), c'est-à-dire le niveau physique et le niveau "DATA LINK", gérant de façon optimale les collisions.

Le client doit toutefois fournir une intervention non négligeable dans les traitements:

- des cas normaux pour la gestion des quatre commandes de base (Appel, libération, émission, réception). L'utilisateur doit préparer ces commandes, analyser les réponses et en fonction de celles-ci, enchaîner la séquence. Les formats tant à l'émission qu'à la réception sont données à la figure 6.5.2.
- des cas d'anomalies soit:
  - à l'émission, s'il y a un nombre trop élevé de collisions de messages.
  - à la réception pour tous les messages s'avérant fautifs et détectés par un CRC invalide.

Ce report vers l'utilisateur de ce travail "fastidieux" et particulier du traitement des anomalies, doublé du fait que Ethernet (comme X21) se différencie nettement de X25, nous conduit à envisager pour ces deux technologies, un protocole de recouvrement d'erreurs (type BSC par exemple) sur la base d'un système de détection d'erreurs (parité, CRC...)



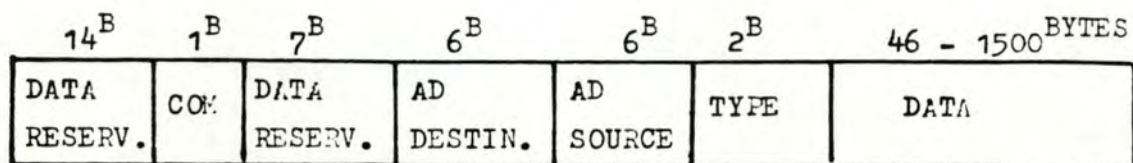


— Fig. 6.5.1 STRUCTURE OF THE DATA LINK PROCEDURAL MODEL. —

L'alignement des réseaux "pauvres" (X21, Ethernet) sur X25 devient alors possible. Ce protocole se situera dans les deux cas au niveau 3 et sera relié aux deux sous-couches du réseau NLD (voir fig. 6.3.3.1). Il assure lui-même la liaison banalisée avec les couches supérieures inamovibles (y compris le niveau de transport L4) et cela quel que soit le réseau. Pour ces couches, tous les réseaux sont transparents et doivent être fiables.

Nous illustrons à la figure 6.5.3 les relations inter-couches qui existeront entre les niveaux pour un réseau Ethernet.



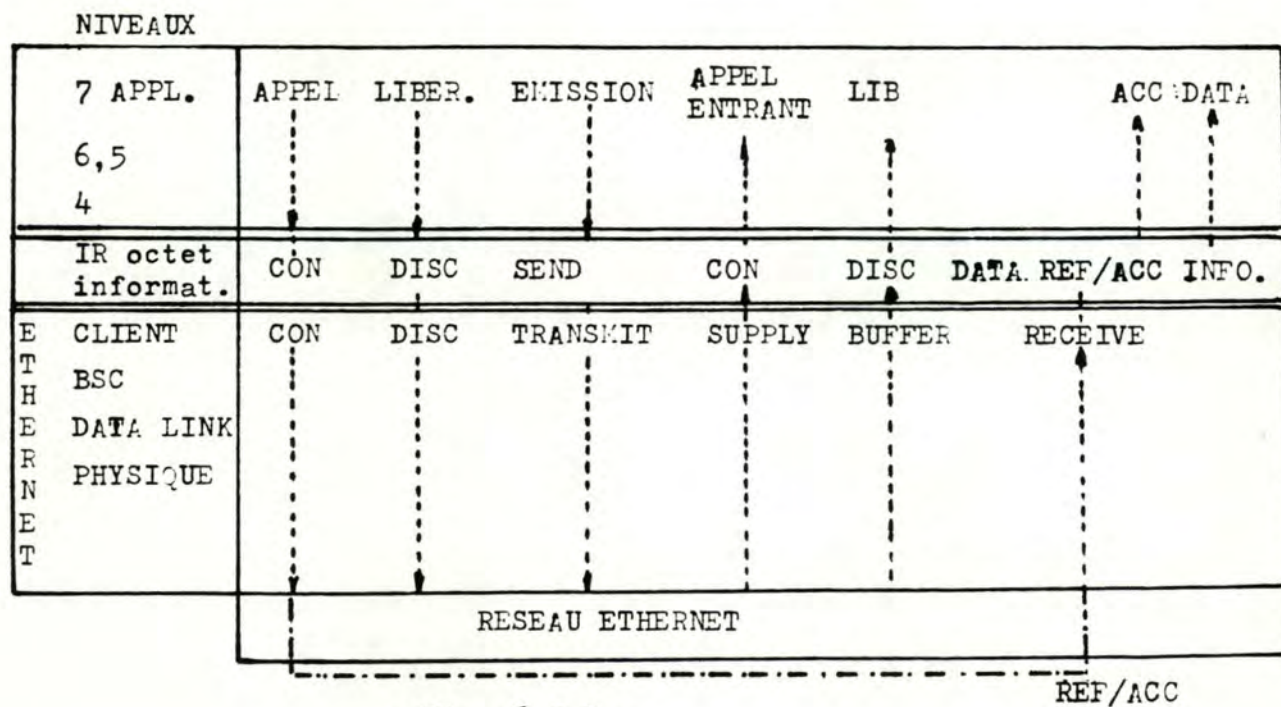


"

- . CON
- . DISC
- . TRANSMIT
- . SUPPLY
- . BUFFER
- (receive)

CLEAR LIBERATION  
INCOMING CALL  
DATA . ACC/REF  
. INFORMATION

- Fig. 6.5.2 -



- Fig. 6.5.3 -

## 6.6 - ALIGNEMENT ASYNCHRONE SUR X25

La procédure normalisée asynchrone s'appuie beaucoup sur la signalisation électrique et nettement moins sur les messages (que la procédure synchrone) pour assurer la séquentialité des échanges.

Or, la politique adoptée et les possibilités très limitées, pour ne pas dire "nulles", dans les contrôles par signaux dans un réseau local, nous conduit à assurer le contrôle des échanges au moyen des messages s'auto-contrôlant tout au long du séquençement.

Dès lors, étant donné que nous avons opté pour l'alignement du mode asynchrone sur X25, nous allons prendre comme modèle ce qui a déjà été réalisé et normalisé pour la liaison terminal asynchrone  $\longleftrightarrow$  RCP via un PAD (ou ADP; module d'assemblage et de désassemblage de paquets).

Nous allons donc résumer le plus succinctement possible les opérations fondamentales, soit:

- les procédures de commandes du PAD par le terminal avec la possibilité de faire appel au protocole X28 déjà cité au paragraphe 5.3.3.4 (voir annexe I).
- la procédure d'accès du terminal au PAD et inversement (voir annexe I )
- la procédure de libération par le terminal ou par le PAD (voir annexe I )
- la procédure du transfert des données (voir annexe I ).

Comme nous l'avons fait pour les autres réseaux soit synchrone (X21) et mixte (Ethernet), nous procéderons à l'alignement de la procédure asynchrone déjà apparentée à X25 par le PAD, grâce aux deux démarches suivantes:



- Nous allons, à partir de ce qui a été envisagé pour X25, examiner les simplifications qui devront être opérées lorsque les relations appelant/ appelé se feront par le support direct d'un réseau élémentaire ou transparent (par exemple, un réseau local).

Comme pour la procédure synchrone, on peut obtenir un tableau des messages qui seront retenus pour d'une part, l'appel et la libération et d'autre part, les échanges de données.

Nous obtenons ainsi la table ci-dessous qui indique que les primitives essentielles sont identiques à celles sélectionnées et codées (pour les autres procédures) dans les octets d'informations entrant (E) et sortant (S).

	Appelant	Appelé
A. Appel	S	E
Appel Acc.	E	S
Rej.	E	S
B. Demande libération	S	E
	E	S
Confirmation libération		
indication libération		pas envisagé
C. Données	S	E
	E	S
D. Réinitialisation		
		pas envisagé
E. Reprise		

- Nous introduirons dans les couches inférieures le protocole BSC pour enrichir cette procédure et l'amener au niveau minimal pour l'assimiler au X25.

## 6.7 - ALIGNEMENT DES PROTOCOLES

Pour les réseaux locaux, qu'ils soient HDLC (X25 ou non), X21 ou Ethernet:

- il n'y a aucune liaison physique permanente ou virtuelle entre les correspondants. La liaison entre deux stations consiste à indiquer à tout moment aux deux stations, qu'elle est appelant ou qu'elle est appelée. Il n'y a aucune liaison physique établie, et de ce fait, l'adresse du destinataire et parfois même de l'appelant, devra se trouver dans chaque message de façon à sélectionner la bonne station appelée.
- lorsque le réseau est couvert par un protocole, par exemple HDLC, celui-ci se charge de délivrer un message valide du point de vue des informations et du séquençement. Il n'en est pas de même pour les réseaux X21 et Ethernet où il n'existe pas de protocole de recouvrement d'erreurs, de sorte que l'utilisateur doit l'envisager de lui-même au niveau 7 de l'utilisateur. Cette façon de reporter le travail de recouvrement d'erreurs à l'utilisateur n'est pas rationnelle et une solution logique et moins dispendieuse sera d'introduire un protocole de contrôle de validité des données.

Toutefois, la mise en oeuvre de ce protocole doit tenir compte qu'une différence importante existe entre X21 et Ethernet.

Pour Ethernet:

- En émission, un message sortant reçoit un "TRANSIT STATUS" lui signifiant si l'émission a été correcte (transmit-OK) où si elle n'a pu se faire ("excessive collision essor" ou  $\overline{OK}$ ).



- En réception, un message entrant sera accompagné d'un "RECEIVE STATUS" comportant: receive OK, frame check error (erreur à la transmission =  $\overline{OK}$ ) et alignment error (le message est amputé en longueur) . Ces indications seront retransmises à l'émetteur sous forme d'ACK ou de NAK. Celui-ci transformera le OK local et le  $\overline{OK}$  distant en ACK, tandis que le  $\overline{OK}$  local ou le  $\overline{OK}$  distant fournira un NAK.

Sur la base de ces accusés de réception, un protocole de recouvrement sera élaboré (type BSC par exemple) pour pouvoir aligner les réseaux Ethernet et HDLC.

Etant donné:

- qu'il alourdirait le programme d'application si on le situait au niveau 7,
- qu'il ne peut occuper le niveau 4 qui est standard pour tous les types de réseaux,

il devra obligatoirement être implémenté au niveau 3, de façon à se présenter de façon similaire pour tous les réseaux. De cette façon, on pourra envisager une intercouche réseau identique tant inférieure (sous le bus) que supérieure (au-dessus du bus) et adjacente à la couche 4.

Pour X 21:

On peut comme on l'a vu plus haut, contrôler la validité des états et des procédures au moyen des messages échangés, mais on ne pourra contrôler la validité des informations pour autant que:

- un caractère BCC ou CRC soit généré au départ et contrôlé à l'arrivée.
- la présence du STX et du ETB/ETX soit contrôlé.

Dans ces conditions, on peut comme pour Ethernet envisager un protocole complémentaire au niveau 3.

En conclusion, du point de vue du protocole de recouvrement d'erreurs, nous avons vu d'une part qu'un protocole X25 existait pour le réseau local de l'HDLC et d'autre part, qu'il fallait en introduire un pour les réseaux de type X21 (synchrone), X20 (asynchrone) et Ethernet, pour pouvoir réaliser l'alignement de l'ensemble. Nous avons choisi le BSC pour réaliser le recouvrement dans le cas de ces derniers.

La différence entre les deux types de protocoles retenus X25 et BSC porte sur leurs performances respectives.

Compte tenu de la banalisation adoptée, ils apparaissent de façon équivalente pour l'utilisateur. Il ne sera mis au courant de leur limite que lorsque le protocole n'est plus à même de jouer son rôle de récupération des messages erronés.

En fait, l'utilisateur sera au courant de cet incident, tant du côté émetteur que du côté récepteur, lorsqu'il recevra un message (DISC + causes).

En effet, comme le protocole tabulaire BSC nous l'indique "il existe une station maître (appelante) et une station esclave (appelée), le transfert des données pouvant être bidirectionnel". En cas d'appels infructueux ou de messages erronés, des tentatives de recouvrement s'opèreront et à la limite il y aura une libération envoyée vers les couches supérieures (DISC dans l'octet d'information).





Toutefois, pour qu'il ait les vertus d'un protocole performant, BSC devrait comporter également les modules:

- de contrôle de flux pour écouler normalement le trafic lorsqu'il y a différence de vitesse à l'émission, en ligne et à la réception.
- de contrôle de séquençement, qui serait supervisé par le protocole de bout en bout du niveau 4.

Il reste à voir s'il serait vraiment utile d'introduire au niveau 3 la possibilité d'une réinitialisation ou d'une reprise, étant donné que si cette pratique est courante, tant pour le réseau HDLC que pour les utilisateurs, il n'en est pas de même pour ceux qui choisissent X21 ou Ethernet.

## CHAPITRE VII - COMPLEMENTS

=====

### 7.1 - INTRODUCTION

### 7.2 - FONCTIONNEMENT MULTI-PROCESSEURS

#### 7.2.1 - Etat d'un processus

#### 7.2.2 - Fonctionnement général et physique en mode mono-processeur

##### 7.2.2.1 - Principes

##### 7.2.2.2 - Opérations de sortie

##### 7.2.2.3 - Opérations d'entrée

#### 7.2.3 - Fonctionnement général en mode multi- processeurs

##### 7.2.3.1 - Les interruptions

- Interruptions entrantes
  - Hardware
  - Software
- Interruptions sortantes
  - Hardware
  - Software

##### 7.2.3.2 - Plusieurs processeurs à niveaux

- Interruptions
- Instructions spécialisées
- Synchronisation

##### 7.2.3.3. - Cas de plusieurs processus

### 7.3 - SESSION ET ÉCHANGES INTERACTIFS

#### 7.3.1 - Application

#### 7.3.2 - Ensemble des primitives de session interactive

#### 7.3.3 - Diagrammes d'états

#### 7.3.4 - Implémentation des fonctions proposées pour la session

#### 7.3.5 - Interactif sur RCP

### 7.4 - REPRISE ET RÉINITIALISATION

#### 7.4.1 - Reset ou réinitialisation

#### 7.4.2 - Restart ou reprise

### 7.5 - RÉDUCTION DE L'ARCHITECTURE



## CHAPITRE VII

---

### COMPLÉMENTS

#### 7.1 - INTRODUCTION

L'architecture du système qui a été étudiée peut être complétée et enrichie en fonction de certaines exigences. Elle peut également être réduite ou appauvrie en raison de certaines occurrences.

Nous citerons dans le cadre de ces compléments, la possibilité de:

- voir travailler plusieurs processus en même temps, c'est à dire d'envisager plusieurs sessions. Un mécanisme assurant la simultanéité de leurs fonctions devra être envisagé ainsi que l'élaboration d'un moniteur plus performant. Des suggestions seront faites au paragraphe 7.2 au sujet du fonctionnement multiprocesseur, le fonctionnement général en mode mono-processeur y sera rappelé.

Plusieurs sessions distinctes impliquent autant de modules différents qui en principe, devraient comporter d'autres primitives et paramètres. Dans leurs projets de normalisation, les instances internationales ont tendance à maintenir, pour tous les modules, les primitives et paramètres de base, en leur ajoutant des primitives spécifiques aux échanges particuliers. Nous examinerons au paragraphe 7.3 ce qu'a prévu le CCITT à ce sujet, pour une session nouvelle, orientée vers le conversationnel.

- envisager l'incidence sur la couche transport (niveau 4) d'une réinitialisation ou d'une reprise au niveau des segments, ayant pour origine une action équivalente de la part de la couche 3 au niveau des paquets. Ce cas est abordé au paragraphe 7.4.

Nous examinerons dans le cas où une réduction de l'architecture est possible (paragraphe 7.5) à quelles conditions la suppression de une ou plusieurs couches supérieures, ou à la limite de toutes, peut être envisagée.



## 7.2 - FONCTIONNEMENT MULTI-PROCESSEURS

### 7.2.1 - ETAT D'UN PROCESSUS

Pour qu'un processus puisse évoluer, il a besoin de données, de place mémoire,... Ces ressources sont en nombre limité dans un système. Il n'est donc pas possible d'attribuer toutes les ressources nécessaires à un processus dès sa création. On peut alors arriver à une situation où un processus n'est plus en possession des ressources indispensables à son exécution. Il est alors dans l'état "suspendu". Par opposition, un processus qui dispose de toutes les ressources dont il a besoin pour pouvoir exécuter l'instruction suivante est dit dans l'état "actif".

Quand un processus est en attente d'un processeur, il est dit "prêt". Seulement, le moniteur peut décider de lui retirer des ressources supplémentaires pour permettre à d'autres processus de progresser. Ce retrait pourrait entraîner le blocage du processus.

Ces différents états sont représentés schématiquement dans la figure 7.2.1.1.

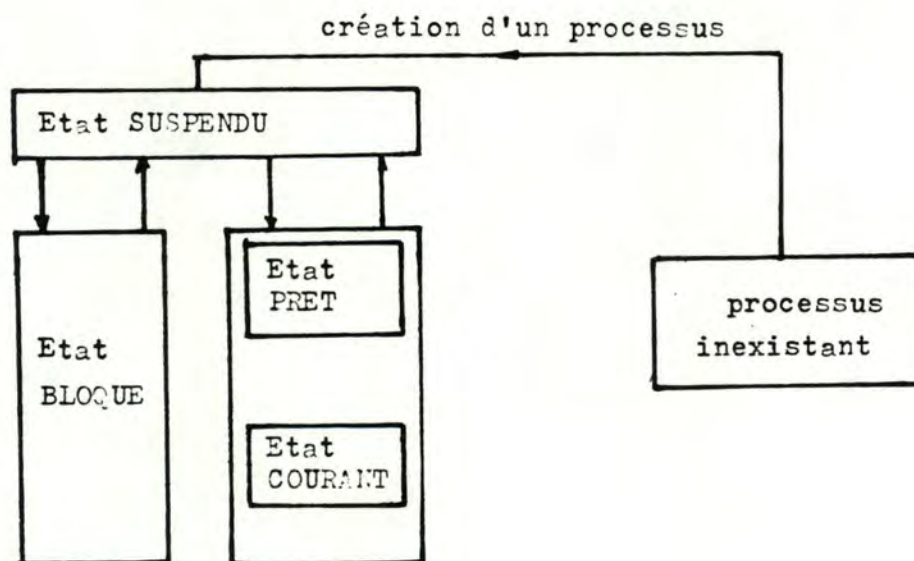


Fig 7.2.1

1. Etat courant: le processus se déroule.
2. Etat prêt: le processus ne peut se dérouler car il lui manque le processeur.
3. Etat bloqué du processus, car la logique l'exige.
4. Etat suspendu: le processus ne peut se dérouler car il lui manque une autre ressource que celle du processeur.

Ils seront codés au moyen de variables binaires v et w, mais on étend cette notion aux niveau d'un processus soit:

vw	
00	1. Etat courant d'un niveau actif (un et un seul).
01	2. Etat bloqué des autres niveaux.
10	3. Etat suspendu du niveau qui deviendra courant lorsqu'un événement se produira.

## 7.2.2 - FONCTIONNEMENT GENERAL ET PHYSIQUE EN MODE MONO-PROCESSEUR

### 7.2.2.1 - Principes

Le fonctionnement est basé sur la scrutation périodique, les interruptions aléatoires et les déroutements vers les routines des différents sous-programmes. Le tout est orchestré par un moniteur de gestion. Reprenons à présent chacun de ces points:



- Scrutation: elle sera effectuée pour tous les programmes qui doivent attendre un certain temps ou une condition, par exemple:
  - du ou des programmes applications en mode "TOP-DOWN" à travers les couches,
  - de la ou des pile(s) de réception via ZL en mode "BOTTOM-UP" à travers les couches L4 à L3.
- Interruptions: les seules interruptions pouvant tout interrompre (même les opérations de scrutation) sont:
  - Interruptions en réception par DMA/HDLC au niveau d'un paquet entrant jusqu'à son dépôt dans la (ou les) pile(s) en entrée au passage de L3 vers L4.
  - Interruption en émission par DMA/HDLC pour transmettre (ZE) vers la ligne. Le moniteur est avisé de la fin de ces opérations (L4 → L3).
  - Les autres ruptures de séquences, (le plus souvent des déroutements) à partir du programme principal sont des JUMP à une adresse de module déterminée avec un retour possible.
- Informations: On distingue 4 types d'informations, ceux-ci sont repris dans le tableau ci-dessous.

<u>Sortie</u>	<u>Entrée</u>	<u>Niveau</u>
a) Commandes appel, lib	réponses appel (acc. rej)	L3
b) Primitives commande/ réponses + param.	primitives réponses/ commandes + paramètre	L4 L5 L6
c) Donnée P. appl. → monit. ligne	donnée lig. monit. → P. appl.	L7
d) Ack. monit. interr.	ack. monit. interr.	

On pourra effectuer plusieurs opérations sur ces informations telles que les tester, effectuer un choix, les transférer, etc...

- + Paramètres: Ils peuvent avoir des origines différentes:
  - Internes: c'est à dire prédéterminés dans une zone mémoire en relation avec le module qui contient la primitive liée à ce(s) paramètre(s).
  - Externes: c'est-à-dire reçus par la station distante (par exemple l'indicatif de l'appelé si la station est appelante).
  - Utilisateur au programme d'application: ils sont introduits le plus aisément au moyen d'un "menu".
- + Dans le fonctionnement général qui suit, seuls les principes chronologiques sont présentés. Pour un examen plus complet du jeu de scrutation, interruptions et sauts avec déroutements, on pourra se référer aux chronologies évoquées dans le traitement des différentes couches (chapitre 5).

#### 7.2.2.2 - Opérations en sortie

La scrutation du programme d'application permet de découvrir soit:

- OPEN-LINK avec adresse destination, adresse prog. appl./adresse session.
- CLOSE-LINK avec Zerr-zone erreur contenant les causes de la libération).
- SEND avec adresse P. A. , longueur, ZE (zone d'écriture) et Zerr (nombre d'essais infructueux).



- RECEIVE avec adresse P.A.  
longueur, ZL (zone de lecture).

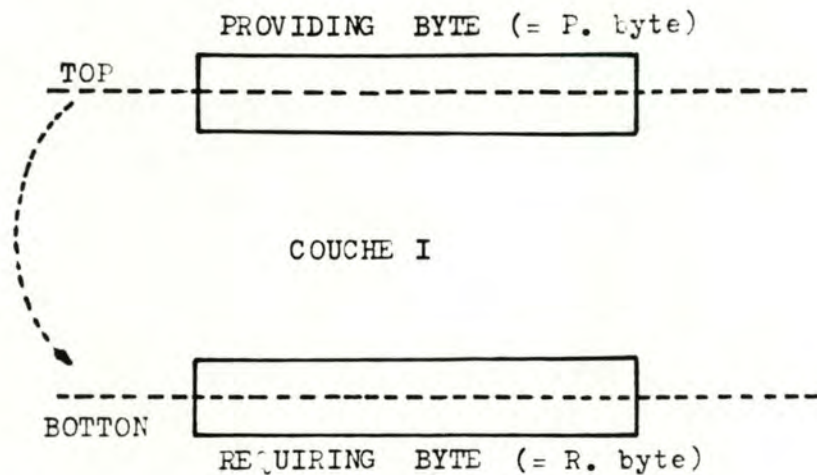


Fig. 7.2.1

Reprenons chaque cas d'une manière un peu plus explicite:

- OPEN génère une primitive d'intercouche CON. Celle-ci traversera les couches de L6 à L3, pour opérer une liaison réseau puis une liaison transport (L4)... D'une manière générale, la progression à travers une couche s'opère de la façon suivante:
  - la primitive du haut positionne le bit correspondant (ici CON) du P. byte (voir fig. 7.2.1).
  - le bit d'acceptation de la primitive devra être positionné dans R.Byte (voir fig. 7.2.1).

- La progression de CON vers le bas ne pourra avoir lieu que si le test d'état du graphe de la couche est correct .

Le moniteur qui contrôlera ces opérations interviendra pour provoquer les opérations de liaison au niveau L3/IR (transfert de l'octet d'interface, transfert des "données utiles de ZE à L4, ou paquet d'appel), puis une opération semblable au niveau L4 (immédiate si la connexion au réseau a déjà été réalisée).

La réponse à cet appel (ACC-REJ avec causes) va d'abord être enregistrée dans la zone mémoire de L4 . Si elle est positive, la séquence se poursuivra aux niveaux L5, L6 puis L7 pour le transfert de données.

- CLOSE: une primitive DISC transitera à travers les couches comme indiqué plus haut, mais en provoquant une remise à zéro (RAZ) de chaque couche.
- Pour l'émission des données (SEND), le mécanisme est similaire. Quand le prog. Application a rempli, (à l'invitation du MONITEUR-MENU), la zone (ZE), le moniteur consulte les niveaux successifs 6, 5, 4 pour introduire leur primitive associée à leurs paramètres, soit:
  - au niveau 6 CDUI
  - au niveau 5 CSUI
  - au niveau 4 TDT (à ce niveau, il y a introduction de N(R) et



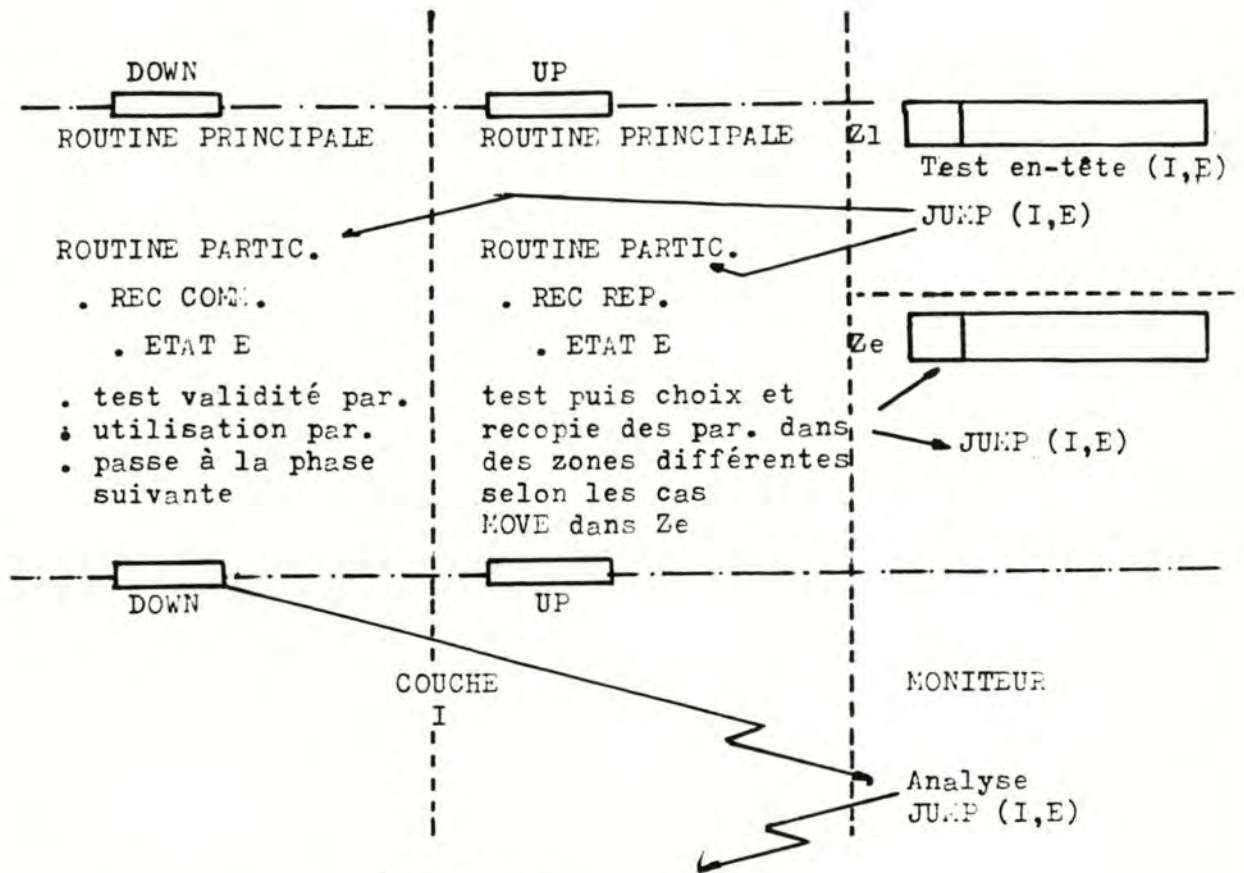
N(S) pour le contrôle du flux).

Ceci doit être accompagné du découpage des blocs en segments qui fourniront des paquets.

- Pour la réception des données (RECEIVE), le mécanisme intercouche opère évidemment dans le sens inverse du SEND.

Au lieu d'introduire des primitives et des paramètres, il s'agira de les analyser et de vérifier leur validité. Ces opérations s'étant révélées correctes, le programme de couche auxquelles elles correspondent, va être activé. Les paquets seront regroupés en segments, puis en blocs, avant d'aller vers le programme d'application.

Ce qu'on vient de voir donne une idée générale des opérations, vu de la part du programme d'application. Il est utile de préciser un peu comment sont articulées les opérations relatives aux couches. Elles sont illustrées à la figure 7.2.2 pour une entrée et une sortie.



- Fig. 7.2.2 -

### 7.2.2.3 - Opérations d'entrée

Le moniteur:

- connaît l'arrivée d'un paquet (interruption DMA/HDLC).
- doit pouvoir faire la distinction entre 1 paquet = 1 segment plusieurs segments.
- doit tester le bit EOT de fin de bloc.
- empile les segments automatiquement à la réception avec la liste des pointeurs.

Par scrutation, les segments sont placés l'un après l'autre dans une zone de lecture ZL pour être analysés.

Le message peut être soit:

- Appel/libération par CALL INCOMING/LIB ou par une réponse CALL ACC/REJ au niveau L3 avec génération de



CON ou DISC vers le (s) couche(s) supérieure (S) concernée(s).

- Une primitive ou sa réponse (avec ses paramètres) d'un niveau 4 (TCR/TCA), 5 (CSS/RSSP- N), 6 (CDS).
- Des données à destination de L7 si N (R/N (S) sont valides. S'ils ne le sont pas, il y aura une réinitialisation.

D'une façon plus détaillée, on aura selon le cas:

- Pour un appel/libération: il s'agira de
  - positionner le bit de l'octet d'interface.
  - activer par "interruption" le moniteur du niveau 3.  
Celui-ci prend (par pointeur) les informations nécessaires.
  - transformer le niveau de ce bit en primitive interne
    - . CON (call incoming/call acc) qui active le TCR de L4
    - . DISC (lib...)
- Pour une primitive: il s'agira de
  - faire un saut dans la routine de la couche I, vérifier la validité et contrôler les paramètres.

- prendre la décision voulue après lecture de la primitive/réponse et des paramètres qui seront utilisés et pris en considération dans une table ou zone consultée selon sa fonction au moment voulu.

† Pour une donnée, il s'agira de

- tester l'en-tête avec saut à la routine pour le contrôle de validité des paramètres de TDT (4) CSUI (5) CDUI (6).

- contrôler N (R)/N (S) et la fenêtre.

- Pour le dernier cas, (ACK), il s'agira de:

- tester le code ACK

- enregistrer N(R) pour recevoir des segments suivants.

### 7.2.3 - FONCTIONNEMENT GENERAL ET PHYSIQUE EN MODE MULTI-PROCESSEURS

#### 7.2.3.1 - Les interruptions

Interruptions entrantes.

- Hardware

Elles sont enregistrées sur un banc d'interruptions. Le câblage de ce banc permet de transformer le niveau de l'interruption en une adresse où va se dérouler le programme.

- Software

- par arrêt et relance du programme, au moyen d'instructions spécialisées du type HALT, WAIT,



INTERRUPT, SUSP (voir plus loin)

- par scrutation de points bien définis. Si le test est positif, il y a arrêt et traitement du cas. Les tests peuvent s'opérer de façon continue ou périodique.
- par le test d'un caractère au passage, avec un saut à une adresse donnée.

Exemple d'application:

ETAT L6 VW	L6	ETAT L5 VW	L5
01	CDS	00	CSS
			1 Après l'opération sortante de CSS, écriture de l'état correspondant à celle-ci ainsi que de ASS dans le même enregistrement
		10	
		00	2 v=0 <u>oui</u> Test <u>message</u> non            entrant
			3 PC: = ASS
			4 Test de vw = 00
			5 Poursuite à l'adresse ASS dans la routine

## Interruptions sortantes

### - Hardware:

voir le cas des interruptions entrantes.

### - Software:

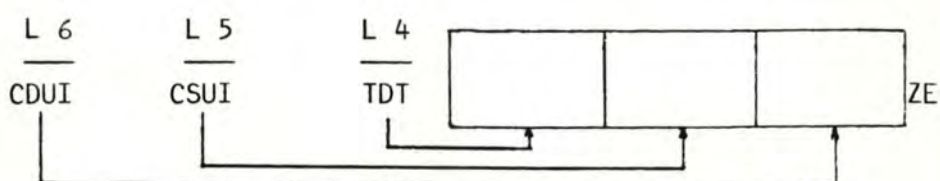
Interruption provoquée (après une opération) par une macro-instruction qui provoquera le déroutement avec sauvegarde, effectue le traitement puis restaure le contexte.

La scrutation d'une série de zones ou de points activera une opération ou un événement.

### Exemple d'application:

il s'agit de remplir une zone ZE de primitives avec leurs paramètres et successivement à partir de niveau différent d'un même processus.

	Etat	
	V = 0 courant	V = 1 suspendu
W = 0 Niveau actif	00	10
W = 1 Niveau inactif	01	11





- a - il existera déjà une cascade de trois interruptions pour passer des niveaux L6 à L4 par le positionnement des variables V et W  
Soit à L4  $W = 0$  (dans zone commune et connue).

TDT      tester au passage dans ZE

$V = 0$     $PC = ASS\ 4$

Test de  $VW = 00$

Poursuite

Retour au moniteur pour passage;

à L5 pour CSUI, avec ASS 5... passage,

à L6 pour CDUI, avec ASS 6...

- b - Quand ZE est rempli, il y a positionnement d'un bit qui sera lu périodiquement et qui provoquera une interruption pour l'envoi du segment contenu dans ZE.

#### 7.2.3.2 - Plusieurs processeurs à niveaux

##### + Interruptions

La technique des interruptions pour le passage d'un niveau à l'autre reste le même que précédemment.

<u>P1</u>	<u>P2</u>	. . .	<u>Pn</u>
L6	L6		L6
L5	L5		L5
- - - - - L4 - - - - -			
L3			

Le changement de processus

$x = 0$  pour processus actifs

$x = 1$  - - - inactifs

se fera par passage d'un jeton ( $x = 0$ ) au processeur qui doit être activé.

Si les processeurs peuvent fonctionner en parallèles, ils peuvent se localiser dans des mémoires différentes. Les échanges de mémoire à mémoire s'opèrent alors par l'intermédiaire du bus commun.

+ Instructions spécialisées (arrêt du programme).

Pour réaliser un programme prenant en compte plusieurs processeurs à niveaux, il faudra prévoir des instructions spécialisées, nous aurons entre autre:

HALT = arrêt complet (reprise par une intervention extérieure).

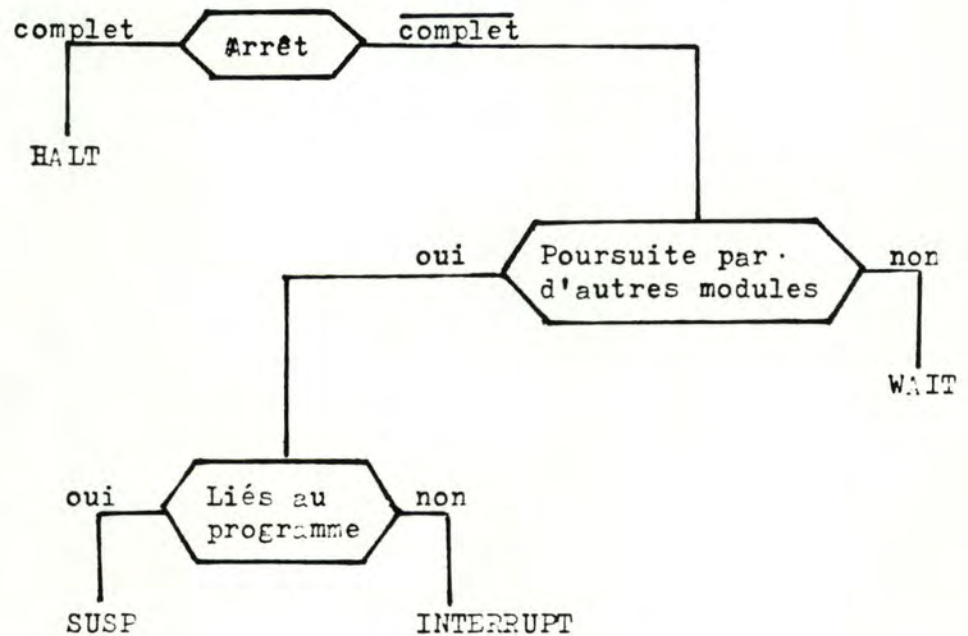
WAIT = attente en un point du programme au moyen d'une boucle (reprise lorsque la boucle teste l'événement attendu).

INTERRUPT =  
arrêt du programme et déroutement vers une routine de traitement (avec sauvegarde et restauration du contexte).

SUSP = arrêt du programme qui se bloquera lorsqu'une opération déterminée surgit (appel opérateur, terminal...) d'autres programmes peuvent continuer à travailler.



Reprenons ces 4 instructions dans un petit organigramme (fig. 7.2.4)



- Fig. 7.2.4 -

#### † Synchronisation

Il existe des moments où il est nécessaire qu'un processus envoie un signal de synchronisation à un autre. Cette synchronisation se fera évidemment alors qu'aucune donnée n'est transférée. Ainsi, un programme peut être placé dans un état de "sommeil" en attendant un signal de synchronisation qui le rend capable de poursuivre.

### 7.2.3.3 - Cas de plusieurs processus

Le mécanisme de fonctionnement global qui vient d'être examiné est relatif aux niveaux d'un seul processus ou programme application. Lorsqu'il y a plusieurs P.A., le moniteur doit disposer d'un module supplémentaire pour réaliser les tâches précédentes en multitraitement et respectant les relations interprocessus qui doivent exister.

Il faudra jumeler la technique d'allocation de tâche par processus et celle des interruptions.

De plus, si l'on désire garder un synchronisme dans les traitements opérés dans chaque processus, il est utile d'adopter en plus une technique de "sémaphore".

Réexaminons rapidement quels seront les grandes particularités de ces multiprocesseurs ou multiprog. Appl.

#### - En\_entrée

En principe, les paquets entrants sont regroupés en segments dans une file d'attente (en réception) de  $\sum ZL$ . En réalité, il s'agit d'une pile de pointeurs vers les différentes ZL.

Ils seront analysés chronologiquement ou selon des priorités à définir. Selon l'adresse source / destination que l'on lit, on connaît le processus et l'on revient au traitement mono-processeur précédemment décrit.



- En\_sortie

C'est par une scrutation "périodique" des prog. Appl. que le moniteur organise les opération sollicitées par ceux-ci soit OPEN-LINK, CLOSE-LINK...

Dans le SEND, il faut distinguer (via l'interface: "NETWORK LAYER" et grâce au moniteur):

- la transmission des primitives des différents niveaux et de leurs paramètres.
- la transmission de données.

Le principe fondamental est que si:

- le Prog. Appl (L7) scruté est demandeur,
- un niveau Li à quelque chose à transmettre,

le moniteur est informé par interruption (ou peut être par scrutation), chaque Pj et Li disposent d'un STATUS qui est examiné régulièrement et testé. Dès lors l'opération de sortie doit être complètement réalisée, autrement on passe au programme suivant.

Le solution dépendra en grande partie de la partition des processus en blocs mémoires autonomes uniquement liés entre eux soit par le bus commun via un contrôleur d'interruption, soit par des messages transmis à une zone commune d'une mémoire ou à des zones individuelles bien déterminées.

### 7.3 - SESSION ET ÉCHANGES INTERACTIFS (ou conversationnel)

Le mode conversationnel est, rappelons-le, le mode d'utilisation d'un système de traitement à partir d'un poste de travail, dans lequel alternent les messages. Après une entrée dans la phase de transfert de données conforme au mode de base simple, une station à le droit de répondre à un bloc d'information par un autre bloc d'information (tenant compte d'acquiescement positif).

#### 7.3.1 - APPLICATIONS

En général, deux types d'applications interactives sont à considérer soit:

- homme - homme
- homme - machine.

Dans ce cadre général, les applications suivantes pourront être reprises (la liste qui suit est loin d'être exhaustive):

- Accès à une base de donnée
- Accès à une directory
- Dialogue entre deux individus utilisant des terminaux teletex
- Videotext
- Fascimilé
- Téléconférence.

#### 7.3.2 - ENSEMBLE DES PRIMITIVES DE SESSION INTERACTIVE

L'ensemble des primitives suivant a été proposé pour une session interactive:

- Data token give
- Date token please
- Interactive data carrier



- Express data carrier
- (in) activity management
- Release token
- Data token accept
- Error recovery
- Interactive capability negotiation
- Quarantine

Ces primitives sont ajoutées aux primitives discutées au paragraphe 5.2, où nous avons les commandes et réponses suivantes:

<u>CSS</u>	- Start	<u>RSSP</u> / <u>RSSN</u>
<u>CSE</u>	- End	<u>RSE</u>
<u>CSA</u>	- Abord	<u>RSAP</u>
<u>CSUI</u>	- Data	<u>RSUI</u> (avec éventuellement TOKEN GIVE PLEASE RELEASE)

<u>CSCC</u>	<u>RCSSP</u>
CSSU	(DT) Appelant/ DT: démarrage d'un timer
CSUR	(DT) Appelé
CSRS	- Suspend
CSRR	- Reactivate
ESUI	- Expedited data
ESQD	
CSQC	Quarantine délivrée et effacée.

Ces primitives doivent bien entendu être accompagnées de leurs paramètres respectifs.

Remarque: il faut noter que les primitives au niveau 4 restent identiques à celles étudiées au paragraphe 5.3.2.

### 7.3.3 - DIAGRAMMES D'ETATS

Les deux figures suivantes 7.3.1 et 7.3.2 représentent respectivement les diagrammes d'états pour le niveau 6 et 5 (présentation / document et session). Les additions faites pour les opérations interactives sont représentées en pointillé.

### 7.3.4 - IMPLEMENTATIONS DES FONCTIONS PROPOSEES POUR LA SESSION

Examinons maintenant les fonctions des primitives ajoutées au niveau session, que l'on retrouve sur la figure 7.3.1.

- |                    |   |
|--------------------|---|
| Token-please:      | - utilise une fonction de demande session dans RSUI sans donnée utilisateur nécessaire.   |
| Token-give:        | <ul style="list-style-type: none"> <li>- utilise un paramètre de CSUI comme jeton.</li> <li>- donne une indication, sans donnée utilisateur.</li> </ul> |
| Data token accept: | - est nécessaire seulement si on désire se protéger contre les erreurs de protocole.  |



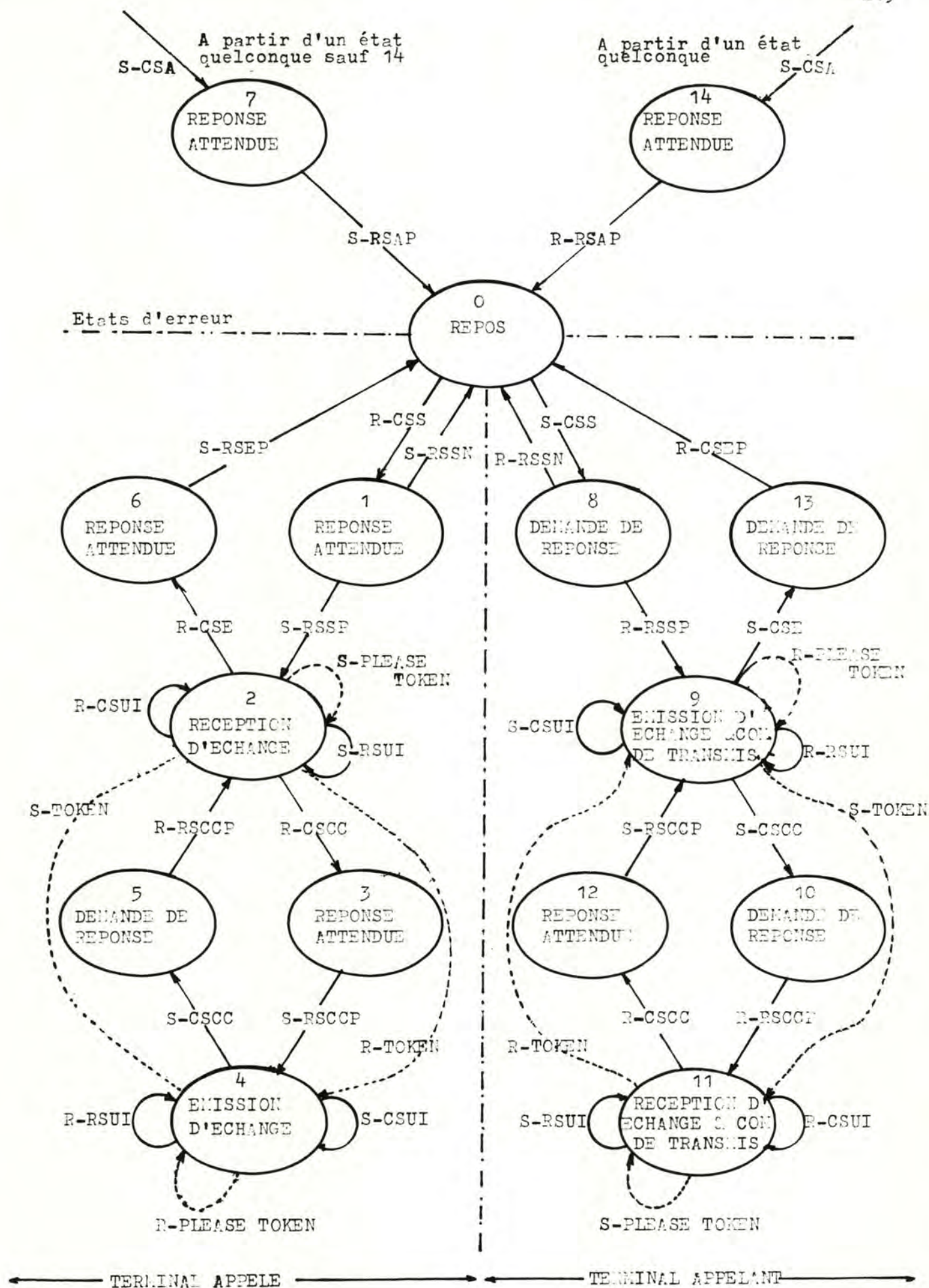


Fig. 7.3.1 Diagramme de transition des états pour une session interactive.

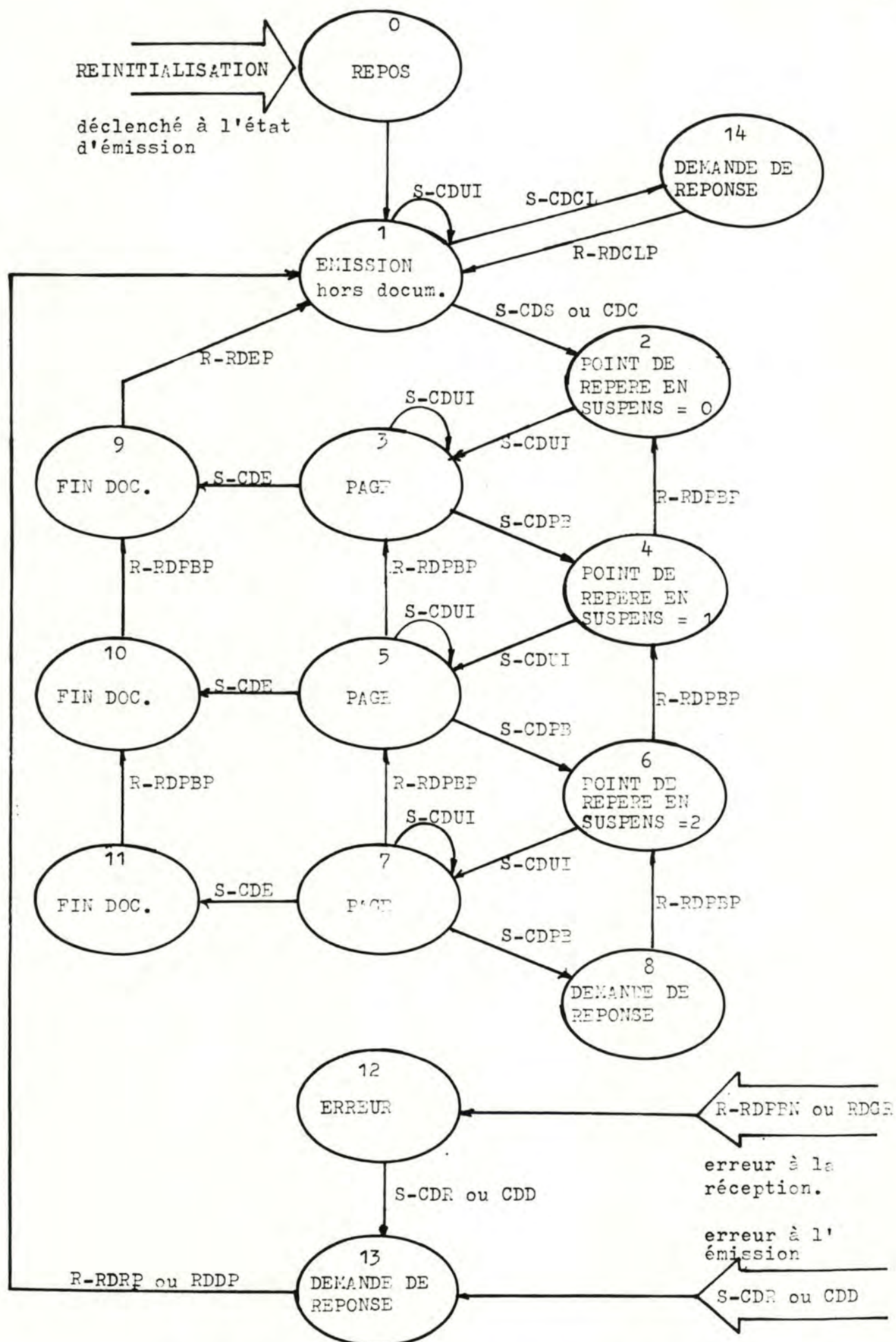


Fig. 7.3.2 Diagramme d'états au niveau document.



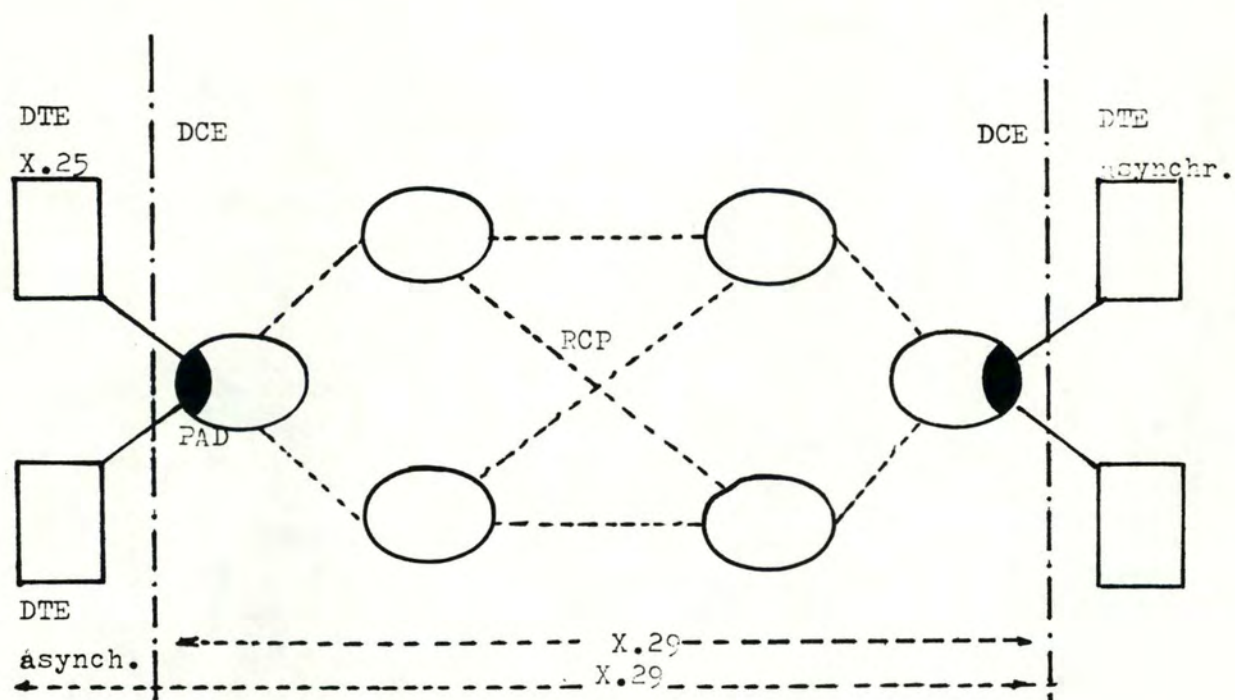
### 7.3.5 - INTERACTIF SUR UN RCP

Parmi les protocoles du réseau RCP (réseau de commutation de paquet), il existe autour du protocole de base X25 de nombreux protocoles utilitaires et supplémentaires. Parmi ceux-ci, le protocole X.29 a pour fonction de gérer les échanges conversationnels de bout en bout interne (RAD-PAD) ou externe (Terminal asynchrone - terminal X25). (voir figure 7.3.3).

X29 est une procédure d'échange de l'information de commandes et des données de l'usage entre un ETTD fonctionnant en mode paquet (ETTD X 25) et un service complémentaire d'assemblage et de désassemblage de paquets (A DP).

Les primitives et leurs paramètres sont complémentaires et inspirés de ceux du X25. Pour les connaître, nous renvoyons le lecteur intéressé à la documentation détaillée X29 (voir annexe J ).

Son intervention est indispensable dans une session interactive, dans le cas de l'interrogation d'une base de données non adaptée à X25.



- Fig. 7.3. -



## 7.4 - RÉINITIALISATION ET REPRISE

Il existe dans le réseau paquet deux primitives "REINITIALISATION" (ou RESET) et "REPRISE" (ou RESTART) utilisées selon la gravité du problème qui peut être interne ou externe au réseau.

Ces opérations ont pour objectif de reprendre le transfert des données à partir du départ, de l'origine et de "gommer" toutes les informations qui se rapportaient à cet intervalle.

Elles font partie intégrante du protocole X25 mais n'existe pas dans les réseaux synchrone (X.21), asynchrone (X.20) et locaux. Il serait d'ailleurs beaucoup trop dispendieux d'introduire la réinitialisation et la reprise dans ces genres de réseaux.

Dès lors, si l'on s'en tient au RCP, une de ces deux fonctions opérées au niveau 3 aura une incidence à l'émission au niveau 4 de transport et même jusqu'au niveau d'application, puisqu'il s'agira de revenir en amont de l'enregistrement.

C'est la raison pour laquelle il est prévu deux primitives d'intercouches RESET et RESTART avec les paramètres nécessaires pour assurer une parfaite resynchronisation de l'opération perturbée.

Examinons succinctement les fonctions principales que ces primitives auront à remplir.

### 7.4.1 - RESET OU REINITIALISATION ( $L_{n-1} \rightarrow L_n$ )

Il s'agit d'une reprise mineure, sans libération, réappel et de plus partielle puisqu'elle ne reprend qu'un fragment du document complet déjà enregistré.

Le point de reprise de ce fragment se situe au niveau:

- a - soit du bloc physique:(exemple: début de l'écran) la garantie d'une bonne reprise n' est pas assurée à tous les coups.
- b - soit en repartant avec le paquet dont la numérotation se situait en début de fenêtre ( $W = \emptyset$ ): la garantie d'une bonne reprise reste aléatoire.
- c - soit au début du niveau du segment précédent en affectant au premier paquet de réinitialisation la valeur  $W = 0$  d'une nouvelle fenêtre. Ceci va obliger le niveau qui reçoit un RESET d'opérer un calcul de repointage du segment. En effet, le contrôle de  $N(R) / N(S)$  du niveau 4 (segment) est indépendant en cours de régime du contrôle  $N(R) / N(S)$  du niveau 3.

Pour la réinitialisation, une resynchronisation parfaite exige une remise à zéro dépendant des fenêtres des deux niveaux.

Rappelons que les paquets déjà enregistrés antérieurement dans le cadre fixé par le retour, seront détruits.

#### 7.4.2 - RESTART OU REPRISE ( $L_{n-1} \rightarrow L_n$ )

Il s'agit d'une reprise totale pour laquelle une libération et un renouvellement de l'appel (en tous cas au niveau 4) est envisagé avec destruction des paquets tandis qu'on effectue la reprise au niveau 6 de présentation (document).



Malgré la reprise à l'origine, cette opération est assez délicate et fait appel à un protocole de contrôle de bout-en-bout.

## 7.5 - RÉDUCTION DE L'ARCHITECTURE

Nous avons vu au chapitre 5, qu'un réseau est un ensemble complexe qui nécessite une structuration permettant de décomposer le système jusqu'à des éléments directement réalisables.

Nous avons utilisé pour ce faire une décomposition en couches. Celle-ci a été proposée par l'ISO (International Standardization Organisation).

L'architecture proposée par l'ISO, comporte 7 couches. Nous rappelons ci-dessous de manière succincte leur fonction:

- couche\_physique: assure le transport de l'information.
- couche\_liaison: assure l'acheminement sans erreurs des blocs d'informations sur la liaison de données.
- couche\_réseau: assure le routage, le contrôle de flux lors de l'acheminement des paquets.
- couche\_transport: assure le séquençement, le contrôle de bout en bout des informations.
- couche\_session: assure la mise en place et le dialogue entre les stations-distances.
- couche\_présentation: assure la bonne présentation des données échangées par les applications.
- couche\_application: assure les fonctions accessibles à "l'utilisateur".

Cette architecture en 7 couches permet une bonne compréhension des différents éléments qui vont intervenir dans la construction d'un système.



Toutes les couches ne sont pas nécessairement utilisées. Par exemple, une simple connexion entre deux ordinateurs identiques pourrait se faire avec uniquement les niveaux 1, 2, 3 et 7.

Examinons maintenant dans quelles conditions les couches supérieures sont nécessaires. Nous donnerons par exemple les conditions pour les couches 6 et 5.

- Présentation:

Cette couche est indispensable lorsqu'une compatibilité doit s'opérer au niveau:

- des formats
- des alphabets de caractères
- de la mise en page...

C'est le cas lorsque les terminaux sont hétérogènes. Dans le cas de terminaux "identiques", on peut être amené à supprimer ou réduire l'importance de cette couche.

- Session:

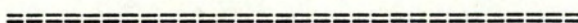
Comme cette couche doit gérer:

- l'échange de texte
- le conversationnel
- l'accès à d'autres réseaux (qu'ils soient à procédure ou à protocole),

l'importance de ce niveau pourra être réduite si on n'utilise qu'une partie de toutes ses possibilités.

Cette réduction nous permet d'envisager une solution minimale, où seuls les niveaux 7, IR, 3, 2 et 1 seront pris en considération. La version minimale sera développée dans le cadre de la maquette.

DEUXIÈME PARTIE





1 - OBJECTIF DU TRAVAIL

2 - HYPOTHÈSES DE TRAVAIL

3 - SPÉCIFICATIONS

1 - INTRODUCTION

2 - APPEL

3 - DONNEES

4 - LIBERATION

5 - TABLES SEQUENTIELLES

4 - CHOIX LIÉS À LA MAQUETTE

5 - INTRODUCTION DES COUCHES 4, 5 ET 6

6 - DISCUSSION DES MODIFICATIONS DE LA MAQUETTE

## 1 - OBJECTIF DU TRAVAIL

---

L'objectif fixé au départ de ce travail pratique est de réaliser une maquette, une implémentation simplifiée des mécanismes d'échange entre deux stations.

L'étude théorique qui a été faite, dans les chapitres précédents, a eu pour objectif d'établir une analyse "complète" et de préparer l'implantation des couches supérieures dans un équipement de transmission de données. Un second aspect de cette étude a été d'étudier la possibilité de permettre aux couches supérieures d'accéder à n'importe quel type de réseaux.

Le travail a été réalisé pour déboucher sur une application concrète dont le support serait un terminal de réseau public ou une station de réseau local. Grâce à ses possibilités de communiquer, de dialoguer à distance de façon normalisée, il pourrait être amené à joindre de nombreux types d'équipements de données distants, greffés sur des réseaux différents, pourvu qu'ils soient normalisés.

La normalisation porte grosso-modo, rappelons-le, sur la structuration en niveaux:

- la couche 4 de transport: pour assurer et orchestrer le fonctionnement des couches supérieures et inférieures indépendamment de l'appelé et du réseau,
- la couche 5 de session: relative à des échanges de texte,
- la couche 6 de présentation: orientée vers la mise en page (alphabets, saut de page, etc...) quel que soit le format d'enregistrement des récepteurs.



La réalisation de tous les logiciels nécessaires, pour l'implémentation du projet d'étude ne peut se faire dans le cadre d'un mémoire d'étudiant. C'est principalement pour cette raison que la concrétisation du travail se limitera à la simulation d'échanges directs entre une application et une autre, ceci exigeant la mise en oeuvre du moniteur et de "l'interface réseau". Il y aura donc relation directe du réseau au niveau 7 d'application (utilisateur) (et vice-versa) en évitant dans un premier temps les couches 4, 5, 6 (celles-ci seront abordées sommairement par la suite).

Remarquons dès à présent que le choix du langage imposera lui aussi des limitations quant à l'implémentation.

Dans une première partie, nous donnerons une description de la solution minimale et des hypothèses y attendant. Ensuite, nous esquisserons les choix concernant l'implémentation et la structuration de ce modèle. Enfin, une discussion des améliorations et des modifications d'hypothèses terminera ce travail.

## 2 - HYPOTHÈSES DE TRAVAIL

---

Nous avons été amené à poser un certain nombre d'hypothèses de travail dans le but de faciliter la réalisation et la présentation de la maquette.

Passons en revue ces diverses hypothèses "simplificatrices".

1 - Une version minimale sera envisagée, nous examinerons ses caractéristiques dans le paragraphe suivant.

2 - Dans le modèle que nous simulerons, on fera l'hypothèse que le travail des trois couches inférieures, à savoir:

- niveau 3 réseau
- niveau 2 liaison
- niveau 1 physique

sera effectué sans problème. Donc, les "données" arrivant au niveau "interface réseau" sont considérées exemptes d'erreurs de transmission.

3 - On suppose que l'enchaînement des opérations entre les deux stations (appelante - appelée) se fait de façon "normale", c'est-à-dire sans réinitialisation, sans reprise, sans interruption, et sans collision lors de la transmission.

4 - La maquette est constituée de deux stations, l'une effectuant les fonctions de l'appelant et l'autre les fonctions de l'appelé. La liaison de données est exploitée en mode bidirectionnel à l'alternat.



5 - Il y a unicité de l'utilisateur au niveau Application. L'échange pour l'appelant et l'appelé seront visualisés sur l'écran des terminaux.

6 - Le langage choisi est le PASCAL UNIX VU - Vrije Universitat - Amsterdam.

Ces diverses hypothèses nous conduisent à réaliser une maquette s'éloignant de la réalité. Un modèle plus réaliste comportant par exemple une gestion d'interruptions multiples, multi-tâche... serait évidemment plus intéressant. Mais étant donné les difficultés que cela poserait, on est amené à restreindre le modèle, entraînant de ce fait une perte de réalisme.

### 3 - SPÉCIFICATIONS

---

#### 1 - INTRODUCTION

---

Soit à simuler un réseau de communications (protocole X25). Nous envisagerons un enchaînement dit "normal" des opérations c'est-à-dire contenant:

- l'appel
- la transmission de données
- la libération.

La maquette est constituée par deux stations, l'une faisant office d'appelant et l'autre d'appelé. On associe à chacune de celles-ci un moniteur chargé de la gérer.

D'autre part, on considère pour chaque poste une version minimale, c'est-à-dire comportant:

- le niveau utilisateur
- l'interface réseau
- le moniteur.

Un schéma général d'une station est donné à la figure suivante (fig. 1).

De plus, dans le cas de la version minimale, seul deux octets d'information seront pris en considération puisque les trois couches supérieures sont court-circuitées. Il s'agit de:

- BOU:information dans le sens descendant
- BIU:information dans le sens montant.

Examinons les différentes phases que l'on peut rencontrer lors de l'établissement d'une communication.



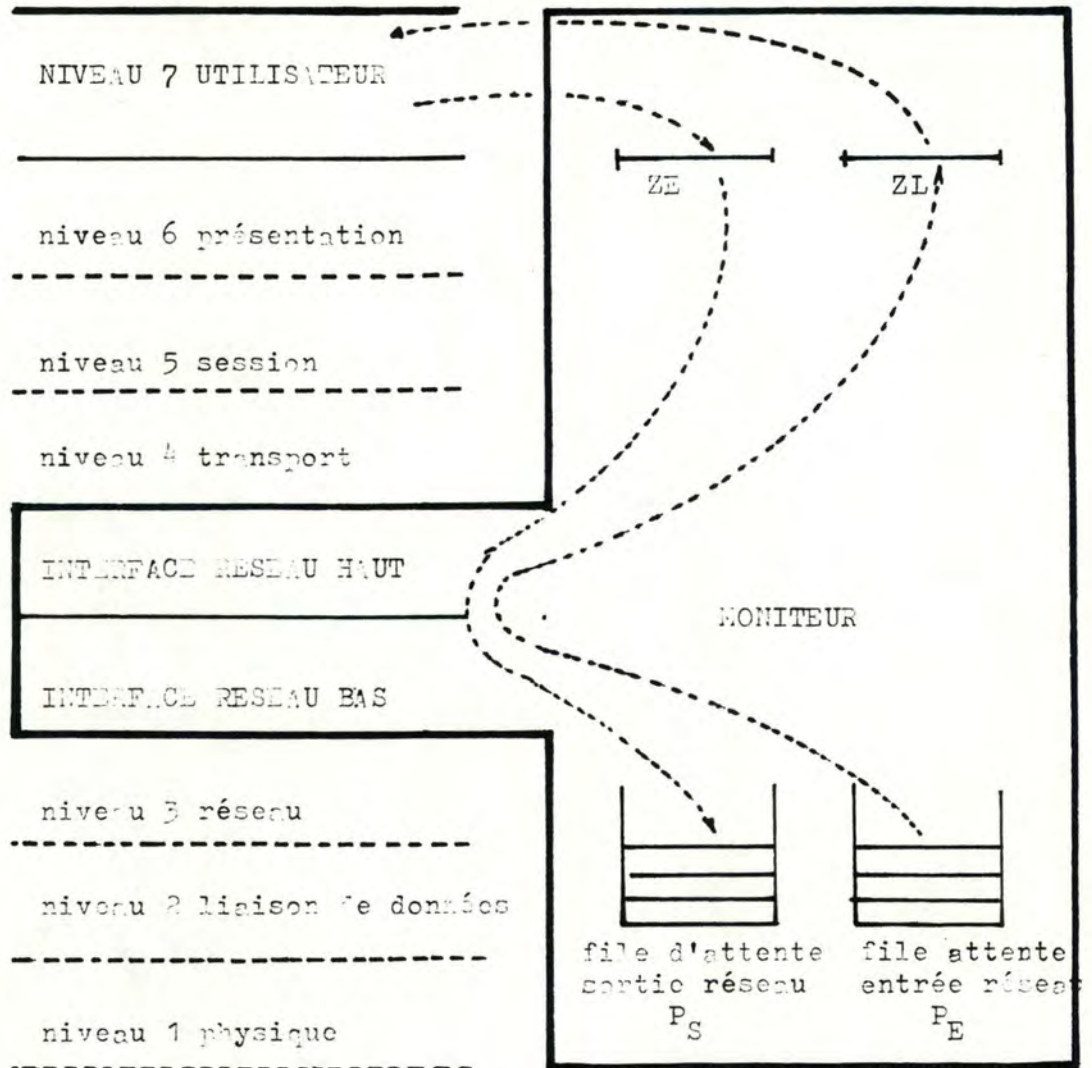


Fig. 1

## 2 - APPEL (voir fig. 2)

1. Un écran d'appel est affiché. Celui-ci demande à l'utilisateur les coordonnées nécessaires afin de réaliser l'appel, à savoir:
  - l'adresse destinataire (obligatoire)
  - l'adresse source (preset)
  - les données d'appel utilisateur (ex. password... facultatif)
  - les facilités (facultatif).
2. Une fois ce menu rempli, le bit - CON - est positionné dans l'octet B0U (a).
3. Le positionnement d'un bit de B0U est détecté par une scrutation périodique du moniteur pour effectuer les opérations suivantes.(b1.)
4. Transfert et formatage du contenu de l'écran dans ZE (zone d'écriture) en fonction de l'opération, avec contrôle du contenu, calcul des longueurs... Ce formatage est réalisé suivant le protocole choisi (c'est-à-dire ici X25) (b2).
5. Dès que le remplissage de ZE est terminé (cela est détecté par une scrutation), l'interface réseau transfère le contenu de Ze vers une pile Ps de sortie (c).
6. Ici deux cas peuvent se présenter:
  - l'appel ne peut aboutir à l'unité (X25: DISC). Dès lors, un reflux provoquera une libération de l'appel par l'introduction d'une signification de libération dans la pile Pe de l'appelant. Cette opération est suivie des phases k, l, m.



# APPEL

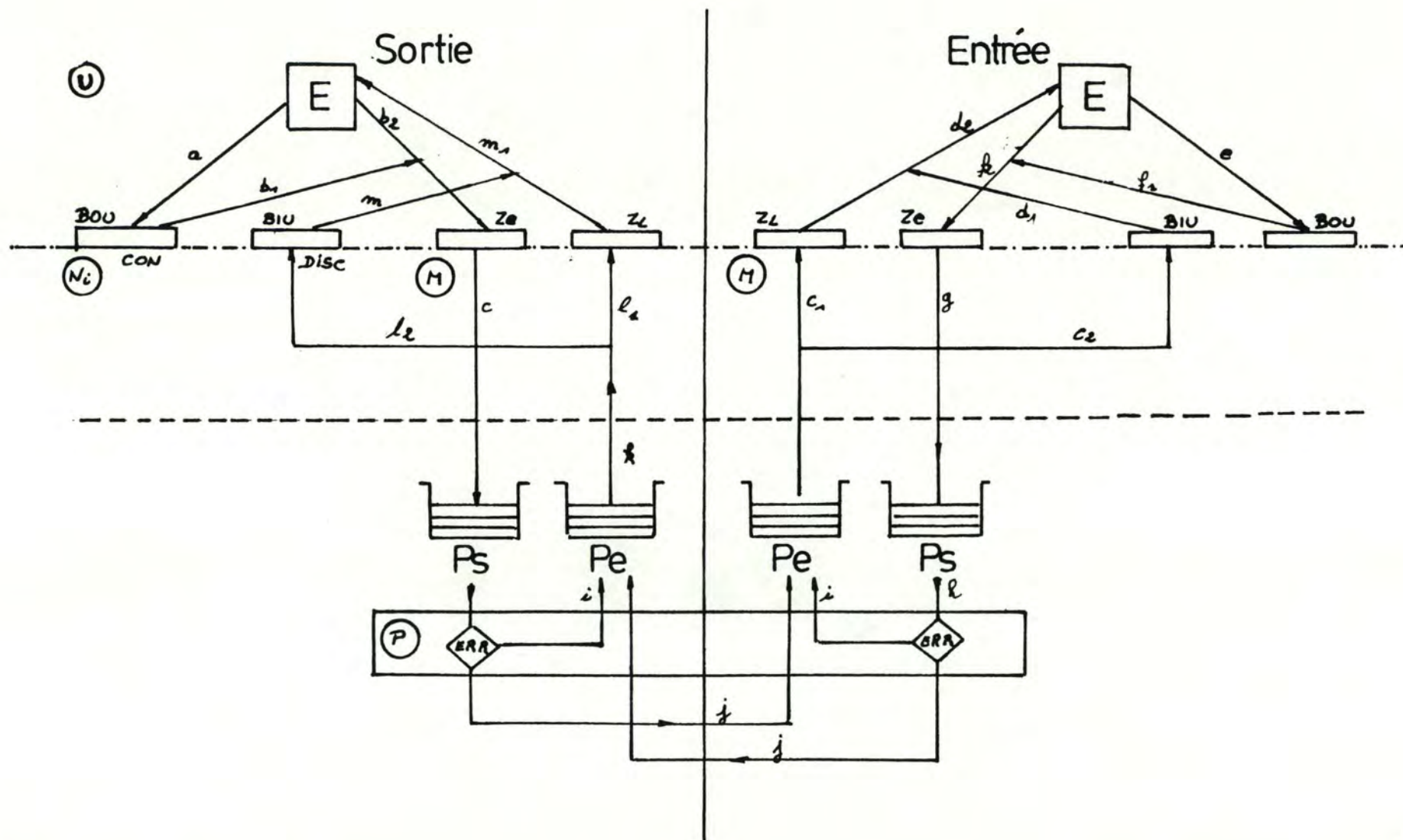


Fig 2. Spécification de l'appel

- l'appel peut aboutir à l'unité appelée.  
Lorsque l'appel réussi, le paquet aboutit dans la pile Pe de l'appelé. L'interface réseau en sera averti et positionnera le bit correspondant à l'opération envisagée dans l'octet BIU.

Le contenu de Pe est simultanément transféré dans ZL (c1 et c2).

Le moniteur est informé de cette opération et pourra alors effectuer un découpage de ZL suivant son contenu (d1 et d2).

7. Une réponse de l'appelé fournira en retour soit un:

- ACC (communication acceptée)
- REJ/LIB (communication refusée).

Ce retour s'effectuera par le cheminement suivant:

- e, f, g, h            pour l'appelé
- k, l, m              pour l'appelant

Remarque: Dans le cadre de la solution minimale, il n'est pas nécessaire de séparer en deux modules les opérations effectuées par l'interface réseau et celles réalisées par le moniteur de la station.

### 3 - DONNEES (fig. 3)

L'appel étant accepté, on peut dès lors effectuer la transmission des données:

- Appelant
  - Inscrire les données à transmettre à l'écran.
  - Positionner le bit - SEND - dans l'octet BOU (a)
  - Le moniteur par scrutation détecte l'opération à réaliser (B 1 ).



# DONNEES

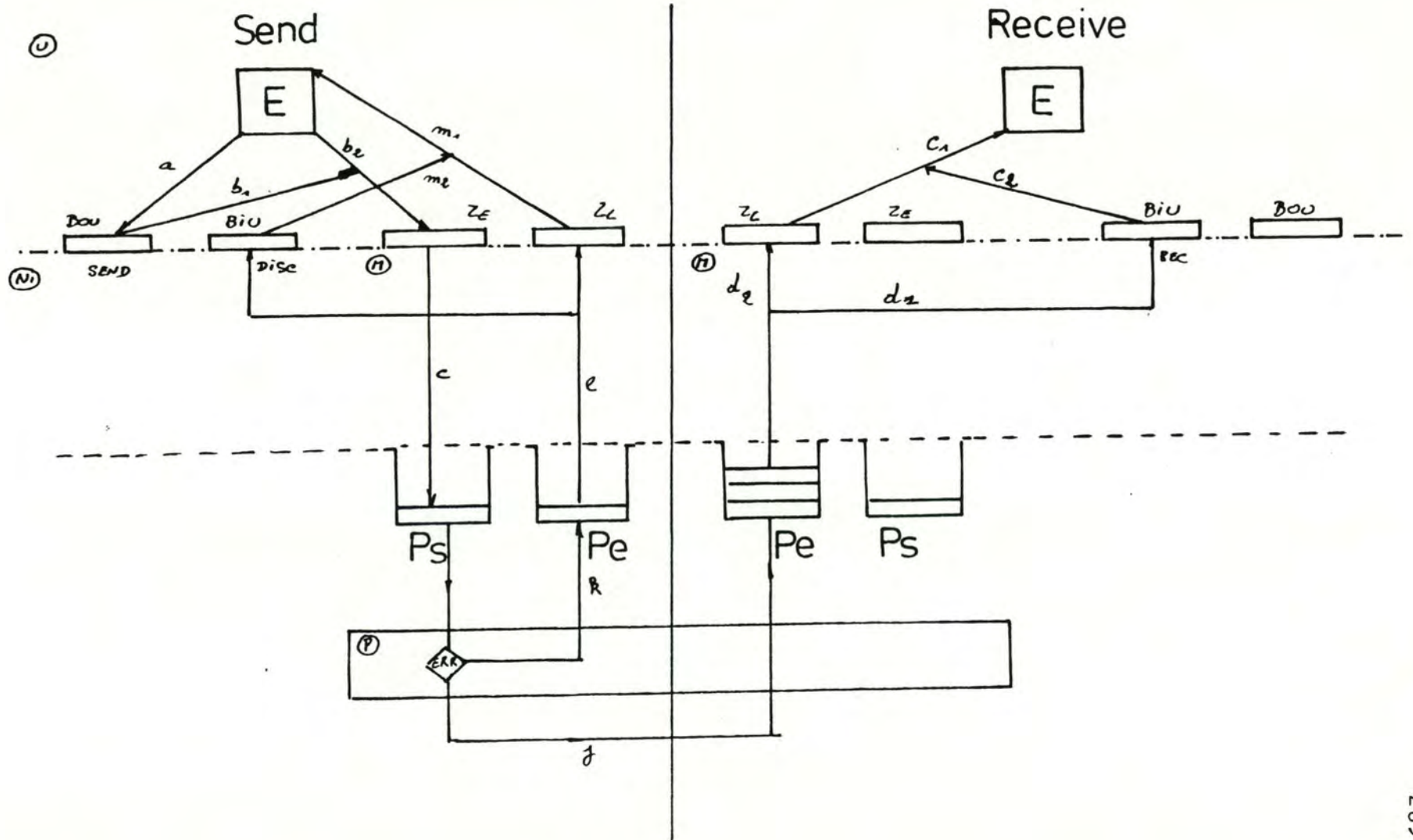


Fig 3. Spécification du transfert de données.

- Il transfere le contenu de l'écran dans ZE (b2) tout en effectuant le formatage correspondant aux données (paquet avec au maximum 128 bytes).
- Le formatage étant terminé, l'interface réseau transfere le contenu de Ze dans la pile Ps de l'appelant (c).
- Ces opérations sont suivies de l'envoi vers le poste appelé des paquets de données ainsi constitués. (j).
- Appelé
  - Du côté appelé, les données sont provisoirement stockées dans la pile Pe de l'appelé.
  - L'interface réseau en est averti, et positionne dans l'octet BIU, le bit - REC - (d1).
  - Parallèlement, le contenu de cette pile est transféré dans ZL (d2).
  - Le moniteur est alors activé, il effectuera le découpage de ZL pour restituer les données à l'écran (c1 c2).

#### 4 - LIBERATION (fig. 4)

On doit distinguer deux cas, suivant que la demande de libération provient directement de l'appelant ou de l'appelé.

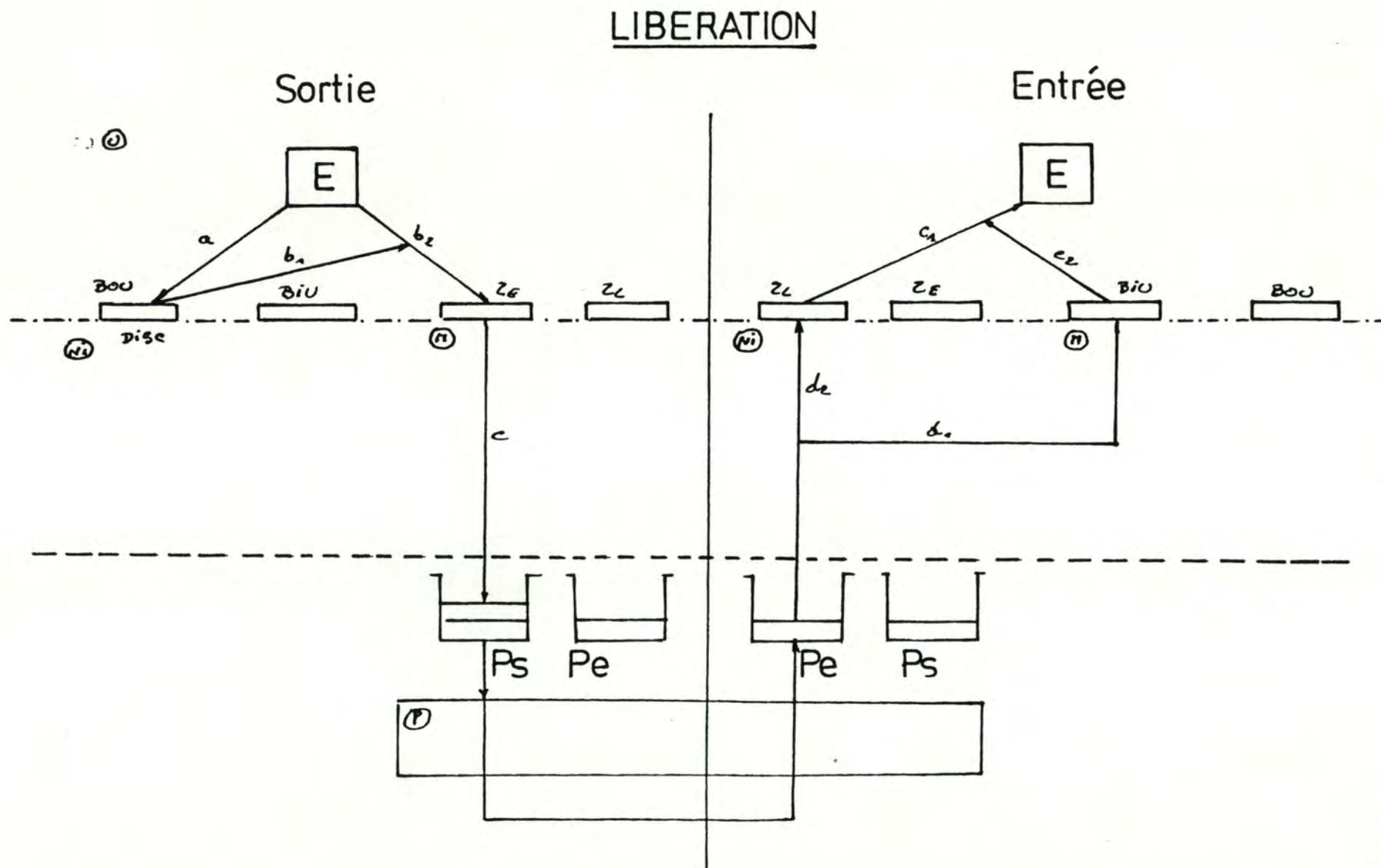
Cas 1. La libération vient de l'appelant.

La procédure de libération, ne peut être lancée qu'à deux moments précis, soit:

- directement après l'appel (établissement de la liaison)
- après le transfert de données.



Fig 4. Spécification de la libération.



- Appelant:

Cette procédure s'effectue de la façon suivante:

- Après avoir fait la demande de libération à l'écran, le bit - DISC - de l'octet B0U est positionné (a).
- Le moniteur averti de cet événement, va effectuer le remplissage de ZE suivant le format retenu en X25 pour le paquet de libération (b).
- Une fois le formatage terminé, l'interface réseau transfère le contenu de la zone d'écriture (ZE) dans la pile PS de l'appelant.(E)
- Ensuite le paquet de libération est envoyé vers le poste appelé.

- Appelé:

- Le paquet aboutit à la pile Pe de l'appelé. L'interface réseau détecte la présence d'un paquet de libération et positionne le bit - DISC - dans l'octet BIU (d1);
- Simultanément le contenu de Pe est transféré dans ZL (d2).
- Le moniteur prend la main ensuite pour effectuer le découpage de ZL et visualiser à l'écran le message utile (c ). Cette phase est traitée plus particulièrement par le programme d'application de la station concernée .

Cas 2. La libération vient de l'appelé.

.....  
Dans ce cas, l'appelant qui va recevoir une demande de libération de la station qu'il a appelée est maître de la transmission.

S'il accepte la demande, la libération s'effectuera comme dans le cas 1.

Dans le cas contraire la communication continue sur sa lancée.



## 5 - TABLES SEQUENTIELLES

Une table d'automate contrôle la séquence des opérations.

Appelant	B inc.	ACC REJ	DISC	LIB prog	
1. App. sort.	-	2 3	3	4	Envoi
2. Données	-	- -	3	4	Send
3. Libération normale	1	- -	-	-	Lib
4. Libération forcée	1	- -	-	-	Lib

Appelé	B	Appel entrant	REC	LIB DISC	
1. Attente	-	2	-	4	
2. ACC/REJ	-	-	3	4	
3. REC	-	-	3/4	-	test dernier bloc 0(4),N(3)
4. LIB	1	-	-	-	RAZ

#### 4 - CHOIX LIÉS À LA MAQUETTE

---

Il est souhaitable que le comportement de la maquette soit le plus proche possible de la "réalité", c'est-à-dire tienne compte du fait que plusieurs programmes puissent tourner en "parallèles".

De fait, quatre programmes (processus) doivent être présents en même temps dans le système. Il s'agit:

- de l'appelant
- du moniteur de l'appelant
- de l'appelé
- du moniteur de l'appelé.

De plus, deux autres problèmes doivent être pris en considération.

- 1 - Les processus, durant leur exécution, doivent s'échanger un ensemble d'informations:
  - de l'appelant vers son moniteur et vice-versa,
  - de l'appelé vers son moniteur et vice-versa,
  - à travers le réseau.

Pour résoudre ce problème de communication inter-processus, on a opté pour la solution qui consiste à faire transiter les messages via un fichier accessible par les deux processus "concurrents".

- 2 - La seconde difficulté découle de la précédente. En effet, il est indispensable d'ajouter une synchronisation entre les processus, ceci afin d'éviter des conflits d'accès qui se traduisent inévitablement par des interférences fatales sur le contenu des fichiers. On est donc amené à utiliser un ensemble de sémaphores (verrous), associés chacun à un fi-



chier. Ceux-ci autorisent (ou non) suivant leur valeur, l'accès aux données communes.

Dès lors, tout accès en écriture ou en lecture doit faire l'objet d'une demande préalable. Ceci se traduit en fait par la lecture du sémaphore associé.

## 5 - INTRODUCTION DES COUCHES 4, 5, 6

---

Dans une première phase, on a court-circuité les trois couches supérieures (transport, session, présentation).

Dans un second temps, il est souhaitable d'enrichir cette première version, par l'introduction de ces trois niveaux, même si cela est fait de façon sommaire.

On envisage, dans ce cadre, un enchaînement dit "normal" des opérations, exactement comme dans la première version. Comme il s'agit d'une introduction, on se limite au dialogue entre les couches au moyen des primitives, leurs paramètres n'étant pas pris en considération.

Examinons les différences que l'on peut rencontrer entre la première et la seconde version.

### A - APPEL

Lorsque la station source reçoit une acceptation à un appel qu'elle a envoyé, une connexion des couches supérieures 4, 5, 6 doit s'opérer de manière chronologique, pour chacun des postes

Notons que l'appelant ne reçoit la confirmation de son appel, que lorsque toutes les couches ont été connectées avec succès.

Il faut remarquer que si l'appelé n'accepte pas la communication, l'appelant en sera averti directement.

La figure 5 représente de façon schématique la séquence de ce dialogue.



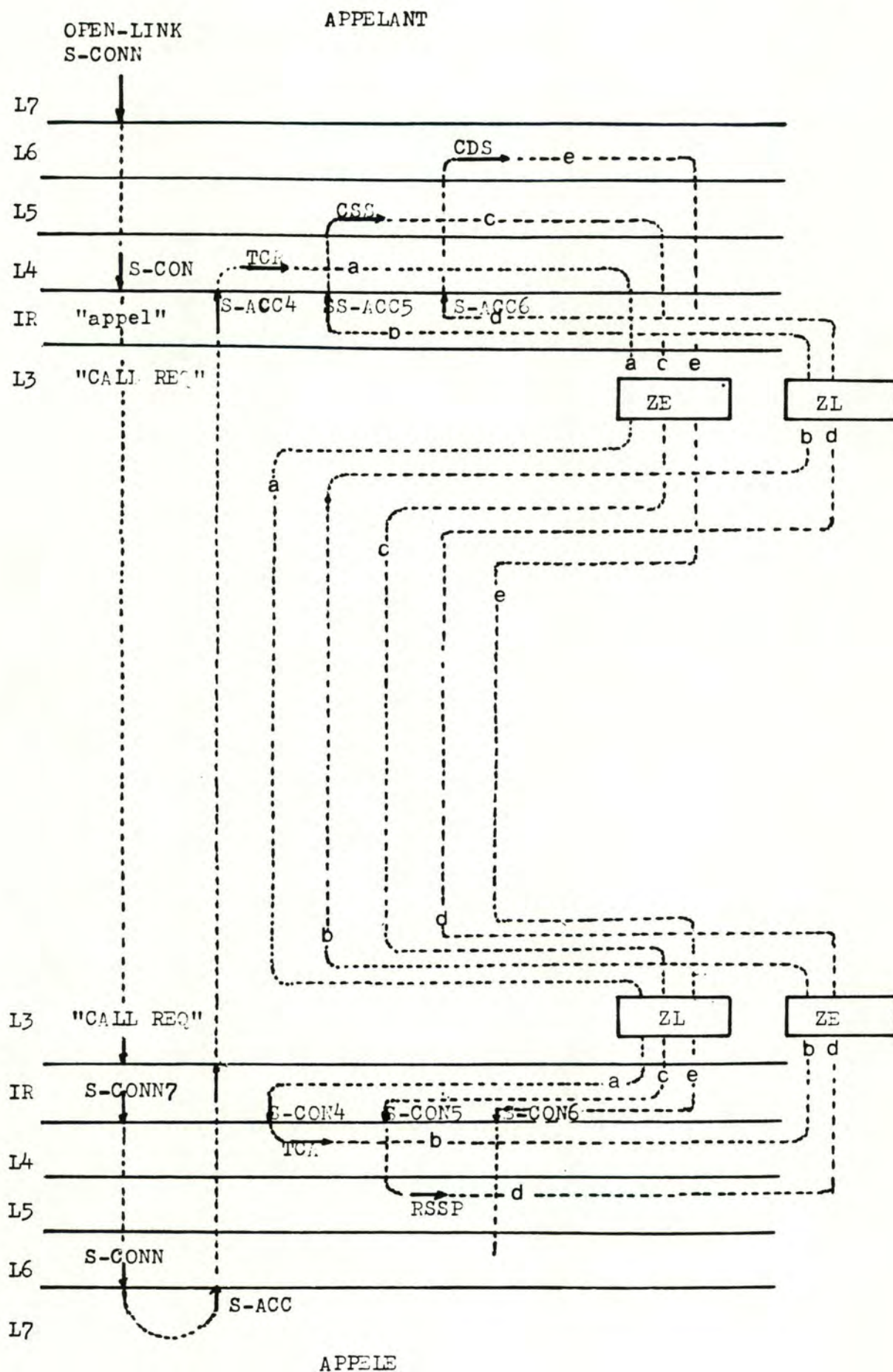


Fig. 5

## B - DONNEES

Dans le cas des données, peu de modifications ont été apportées. On trouvera néanmoins, à la figure 6, un schéma reprenant la chronologie d'un transfert de données.

## C - LIBERATION

A l'inverse de l'appel, une libération va provoquer une déconnexion successive des couches 6, 5, 4, pour finir par l'envoi d'un paquet de libération.

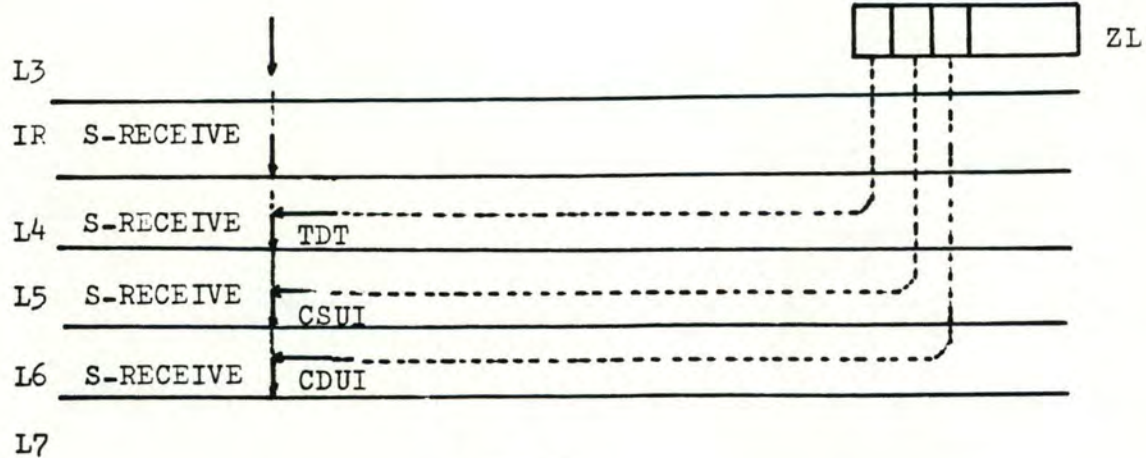
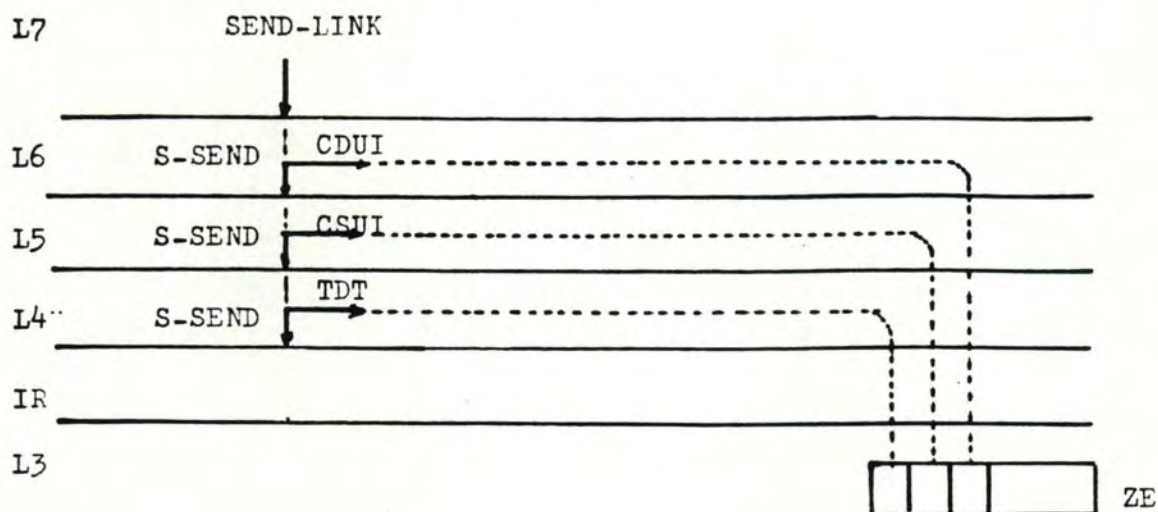
La figure 7 représente schématiquement le dialogue entre les deux postes, lors d'une libération.

L'imbrication des opérations s'effectue suivant la séquence de base illustrée à la figure 8.

Remarque: Les programmes correspondant aux différentes couches sont implémentés de façon séquentielle.

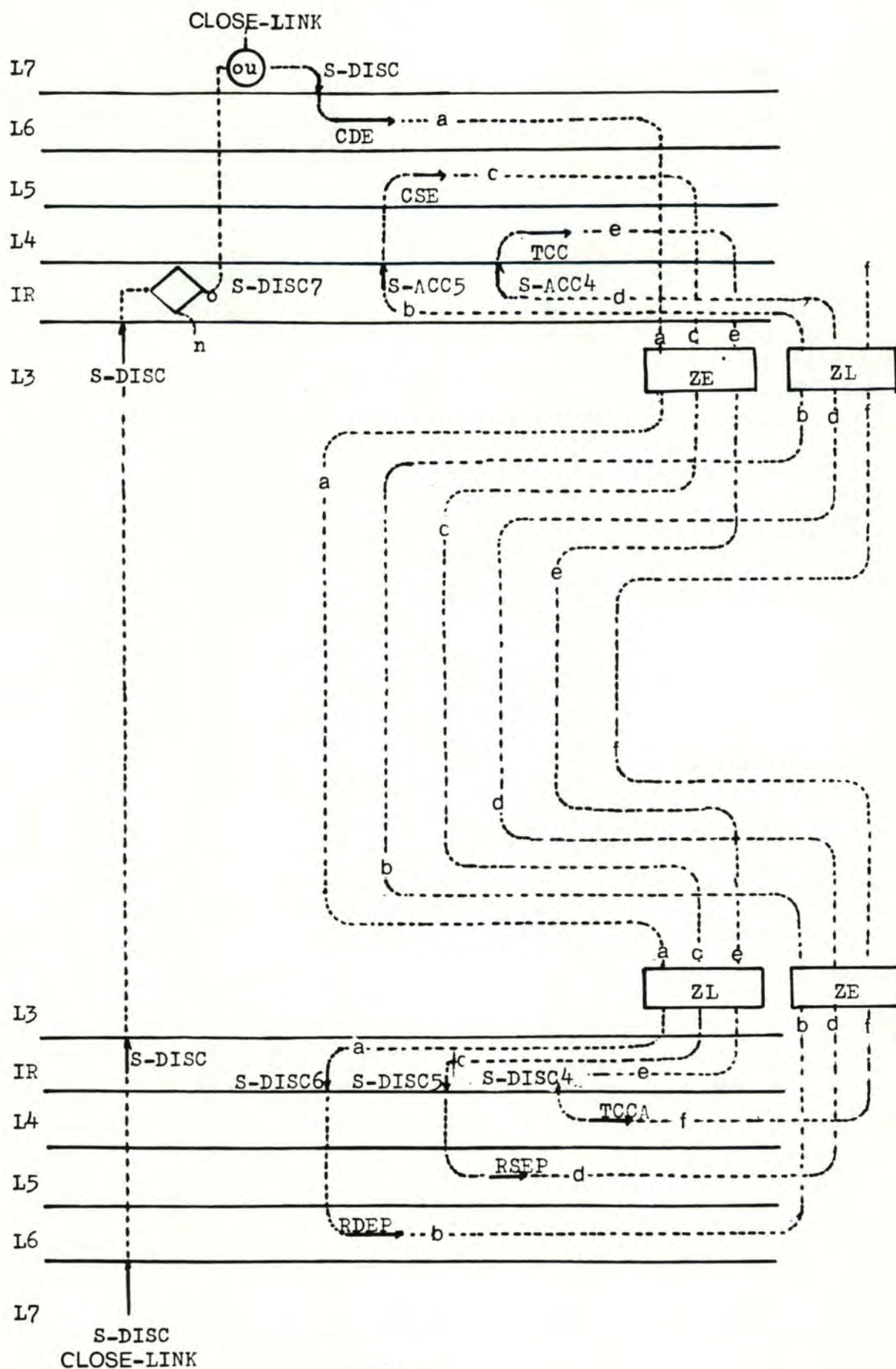


## APPELANT



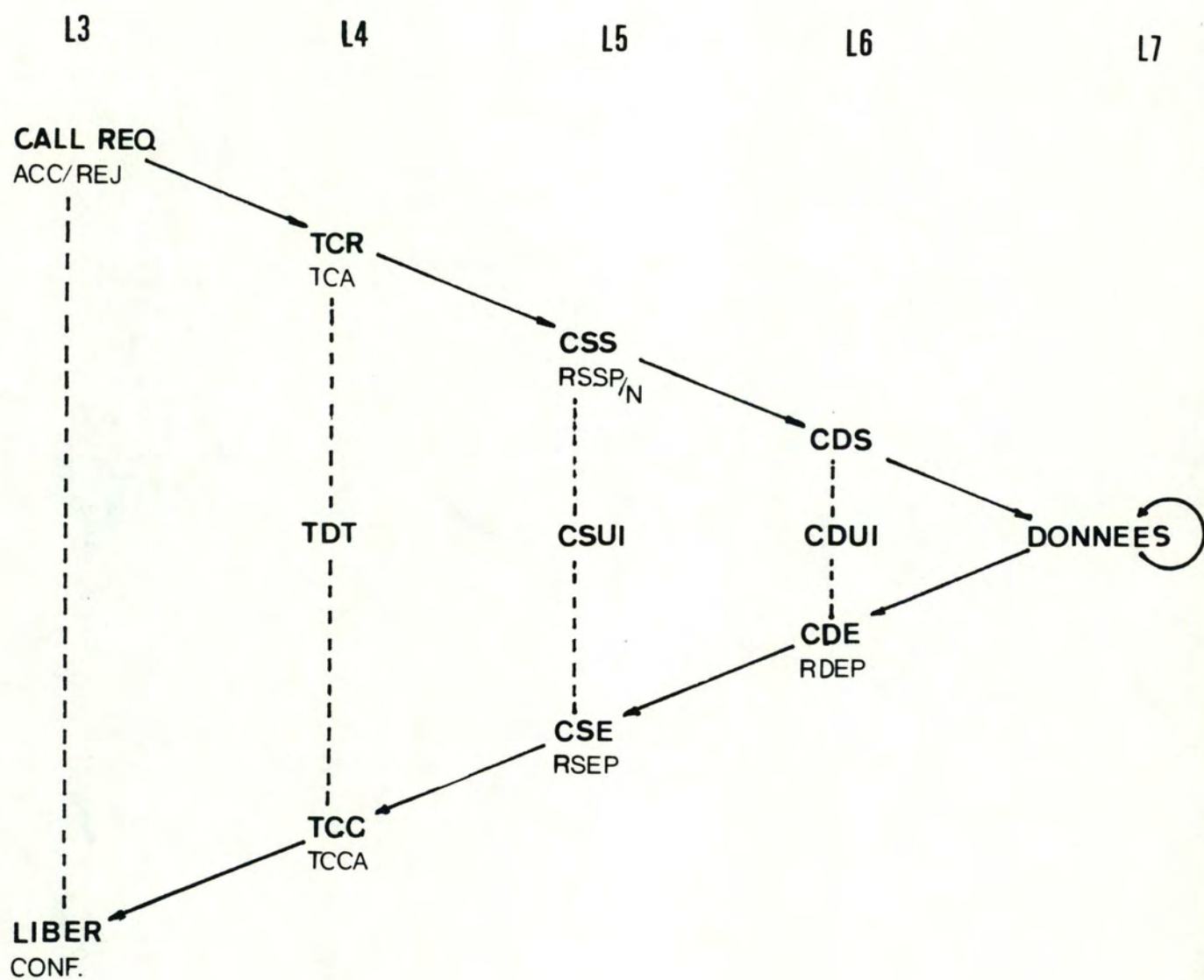
## APPELE

- Fig. 6 -



- Fig. 7 -





— Figure 8. Séquence de base. —

Les primitives sont envoyées d'une station à l'autre, via des paquets dits "de données". On distingue un paquet de données "pures" et un paquet de données représentant une primitive, par le positionnement à 1 du bit 0, qualificateur de données (voir annexe G).

Rappelons qu'un paquet de données a la structure suivante:

octet	8	7	6	5	4	3	2	1
1	1	0	0	1	ADRESSE			
2	DESTINATAIRE							
3	P(R)		M		P(S)		0	
4	PRIMITIVES							

Le quatrième octet représente l'identificateur des primitives. Le codage de celles-ci, pour les commandes et réponses des niveaux 4, 5 et 6, est représenté aux figures 9 et 10.

Afin de retrouver le niveau auquel la primitive est adressée, il serait intéressant d'adopter un mécanisme permettant de mettre en relation une primitive et sa couche. Le codage, tel qu'il est définit par le CCITT, ne permet pas d'adapter simplement un tel mécanisme. Une solution serait de modifier les codes, de manière à se servir des deux premiers bits pour identifier la couche. On pourrait choisir par exemple:

bit				couche
8	7	...	1	
0	0			niveau 4
0	1			niveau 5
1	0			niveau 6



Cette méthode permet de laisser 64 possibilités de codage de primitives pour chaque couche.

Il faut remarquer qu'au niveau 4, pour rester cohérent avec le choix effectué ci-dessus, on est amené à inverser les deux quartets constituant l'identificateur de la primitive. Par exemple, TCR qui était codé 11100000 sera codé par 00001110, ce qui respecte notre choix de solution où les deux premiers bits, au niveau 4, sont 00.

CODAGE DES IDENTIFICATEURS DE COMMANDES ET REPONSES  
AU NIVEAU TRANSPORT

Commande/ Réponse	Numéro de bit							
	8	7	6	5	4	3	2	1
TCR	1	1	1	0	0	0	0	0
TCA	1	1	0	1	0	0	0	0
TCC	1	0	0	0	0	0	0	0
TCCA	1	0	0	0	0	0	0	0
TDT	1	1	1	1	0	0	0	0

CODAGE DES IDENTIFICATEURS DE COMMANDE ET DE REPONSE  
POUR LE NIVEAU SESSION

Commande/ Réponse	Numéro de bit							
	8	7	6	5	4	3	2	1
CSS	0	0	0	0	1	1	0	1
CSE	0	0	0	0	1	0	0	1
CSA	0	0	0	1	1	0	0	1
CSCC	0	0	0	1	0	1	0	1
CSUI	0	0	0	0	0	0	0	1
RSSP	0	0	0	0	1	1	1	0
RSSN	0	0	0	0	1	1	0	0
RSEP	0	0	0	0	1	0	1	0
RSAP	0	0	0	1	1	0	1	0
RSAP	0	0	0	1	1	0	1	0
RSCCP	0	0	0	1	0	1	1	0
RSUI	0	0	0	0	0	0	1	0



CODAGE DES IDENTIFICATEURS DE COMMANDE  
DU NIVEAU PRESENTATION

Commande	Numéro de bit							
	8	7	6	5	4	3	2	1
CDS	0	0	1	0	1	1	0	1
CDC	0	0	0	1	1	1	0	1
CDE	0	0	1	0	1	0	0	1
CDR	0	0	0	1	1	0	0	1
CDD	0	0	1	1	1	0	0	1
CDPB	0	0	1	1	0	0	0	1
CDRP	0	0	1	0	0	0	0	1
CDRR	0	0	1	0	0	1	0	1
CDCL	0	0	1	1	1	1	0	1
CDUI	0	0	0	0	0	0	0	1

CODAGE DES IDENTIFICATEURS DES REPONSES  
DU NIVEAU PRESENTATION

Réponse	Numéro de bit							
	8	7	6	5	4	3	2	1
RDEP	0	0	1	0	1	0	1	0
RDRP	0	0	0	1	1	0	1	0
RDDP	0	0	1	1	1	0	1	0
RDPBP	0	0	1	1	0	0	1	0
RDPBN	0	0	1	1	0	0	0	0
RDCLP	0	0	1	1	1	1	1	0
RDGR	0	0	0	0	0	0	0	0
RDRPN	0	0	1	0	0	0	0	0
RDRPR	0	0	1	0	0	1	0	0

## 6 - DISCUSSION DES MODIFICATIONS DE LA MAQUETTE

---

De par la structure de la maquette, un changement de protocole nécessiterait un nombre limité de modifications. Pour ce faire, il serait nécessaire d'apporter diverses transformations sur les procédures réalisant la construction et l'analyse des paquets de format X25 afin de les rendre cohérentes avec d'autres protocoles (Ethernet...)

Comme vu dans le paragraphe précédent, les couches sont implémentées par des processus "séquentiels". Il serait intéressant qu'elles soient implémentées par des processus "parallèles".

D'autre part, on pourrait étoffer la simulation en implémentant les cas suivants:

- a - Communication entre une station appelante et x stations appelées - Transfert unidirectionnel.
- b - Communication entre une station appelante et x stations appelées - Transfert bidirectionnel - Renversement du sens de l'échange.
- c - Communication entre une station appelante et x stations appelées mais en faisant intervenir progressivement les fonctions essentielles du niveau 4 telles que:
  - contrôle de bout en bout
  - contrôle de flux
  - séquencement
  - ...
- d - Communication entre une station appelante et x stations appelées avec l'ensemble des couches (4, 5 et 6). La couche 5 comportera plusieurs processus (Télex, conversationnel, processus industriel...).



## C O N C L U S I O N S

=====

L'objectif de ce mémoire était d'une part d'étudier les couches supérieures de l'ISO et la possibilité de les rendre compatibles avec différents réseaux et d'autre part, de développer un projet d'implémentation.

Pour arriver à ce résultat, nous avons été amené à examiner:

- les couches supérieures de l'ISO (4, 5, 6, 7)
- la progression de primitives dans les couches supérieures
- un interface "banalisé" (alignement des procédures et protocoles les plus faibles sur X.25).

Comme on l'a vu, ce problème est vaste et donc loin d'être complètement résolu. Il reste en effet bon nombre de sujets à résoudre, parmi lesquels nous pouvons citer par exemple:

- l'introduction d'une session autre que la transmission de texte et notamment le processus industriel, car celui-ci manque à l'heure actuelle d'une normalisation.
- la multiplicité des processeurs qui nécessite un moniteur réentrant.
- ...

En ce qui concerne la seconde partie, on pourrait prolonger l'étude dans les directions suivantes (voir paragraphe 6, modification de la maquette):

- multiplication du nombre des stations appelantes et appelées.
- simulations complémentaires correspondant à différentes sessions.



## B I B L I O G R A P H I E

### BELL TELEPHONE MANUFACTURING COMPANY

Kommunicatie funkties

Proposal transport service

BTMC 17.02.81

### CCITT

Projet avis S 62

AP VII N° 78 - F

### CCITT

Procedures for the exchange of control information and user data between a packet assembly/disassembly facility (PAD) and a packet mode DTE or another PAD.  
CCITT fascicle VIII 2 - Rec. X.29

### COMPUTER NETWORK

R. Desjardins

Overview and status of the ISO reference model of open systems interconnection.

Computer networks 5 1981.

### COMPUTER NETWORKS

ISO

Data processing-open system interconnection

Basic reference model

Computer networks 5 1981.

## COMPUTER NETWORKS

H.J. BURKKARDT, S. SCHINDLER

Structuring Principles of the communication  
Architecture of open systems A systematic  
approach.

Computer networks 5 1981

## EA:

A. SABATIER (4)

- Le multibus et ses signaux

EAI 263 15.02.79

- L'utilisation du multibus

EAI 264 01.03.79

- Le SBC 30/30 et sa mémoire RAM à double accès

EAI 266 01.04.79

- La carte "contrôleur intelligent" de communication

EAI 267 15.04.79

- le SBC 86/12, un micro ordinateur 16 bits sur une  
seule carte

EAI 270 01.06.79

- Le moniteur temps réel multitâche RMX-80

EAI 272 15.09.79

## ECMA-72

Standard ECMA-72

Transport protocol

Janvier 1981

## ECMA

A. BOURGIGNON

Proposed tools for the definition of the session  
protocol

ECMA ITC 23/80/98 p. 1-11

Juin 1980.



## ECMA

Session protocol

ECMA ITC 23/81/100 p. 1-120

Juillet 1981

## ECMA

THIRIOT ET BOURGIGNON

Formal description of the ECMA session protocol

ECMA ITC 23/81/91 p. 1-19

## ECMA

A. BOURGIGNON

State diagrams for the session protocol machine

ECMA ITC 23/80/144 p. 1-9

## ECMA

Data presentation protocol

ECMA ITC 23/81/141

## INFORMATIQUE INDUSTRIELLE

A. SABATIER

Une structure modulaire autour d'un "micro"  
pour le contrôle-commande d'un processus

Juin - Juillet 79

## INTEL

iSBC 80/20, iSBC 80/20-4

Single board computer

INTEL

Ethernet communications controller  
intel

C. MACCHY et J.F. GUILBERT

Téléinformatique  
Dunod informatique  
1979

G. PUJOLLE

La télématique: réseaux et applications  
Eyrolles  
1982

SOBEMAP

La communication inter-programme  
Sobemap.

TRANSPAC

Accès au service de circuit virtuel TRANSPAC  
Réseau de transmission de données par paquet  
Direction générale des télécommunications.

TANENBAUM (81)

Computer networks  
Prentice Hall  
1981

MEYER Jean-François

Etude comparative d'architectures de réseau par  
rapport au modèle de référence OSI de l'ISO.  
Mémoire présenté en 1982 (FNDP Namur).



FACULTES  
UNIVERSITAIRES  
N.D. DE LA PAIX

**NAMUR**



---

INSTITUT D'INFORMATIQUE

ANNEXES

LES COUCHES SUPERIEURES DE L'ISO

COMPATIBILITÉ AVEC DIFFÉRENTS  
TYPES DE RÉSEAU

PROJET D'IMPLÉMENTATION

PROMOTEUR: J. BRUNIN

MÉMOIRE PRÉSENTÉ PAR  
HENRION PHILIPPE  
WALTZING MICHEL  
EN VUE DE L'OBTENTION DU GRADE DE  
LICENCE ET MAÎTRE EN INFORMATIQUE

ANNÉE ACADEMIQUE 1982-1983

## TABLE DES MATIÈRES DES ANNEXES

### ANNEXE A:

DESCRIPTION GÉNÉRALE ET RÈGLES DE FONCTIONNEMENT APPLICABLES AUX DIAGRAMMES DES ÉTATS, . . . . .	1
---	---

### ANNEXE B:

GRAPHES D'ÉTATS DU NIVEAU TRANSPORT, . . . . .	4
--	---

### ANNEXE C:

GRAPHES D'ÉTATS DU NIVEAU SESSION, . . . . .	17
--	----

### ANNEXE D:

FONCTION DE PRÉSENTATION DU CODE ASCII ET ASCII ÉTENDU, . . . . .	27
---	----

### ANNEXE E:

PRIMITIVES ET DIAGRAMMES D'ÉTATS DU NIVEAU PRÉSENTATION, . . . . .	47
--	----

### ANNEXE F:

PROTOCOLE D'APPAREIL VIRTUEL, . . . . .	57
---	----

### ANNEXE G:

PROTOCOLE STANDARD X.25, . . . . .	71
------------------------------------	----

### ANNEXE H:

LA PROCÉDURE X.21 ET SON ALIGNEMENT SUR X.25, . . . . .	81
---	----

### ANNEXE I:

ASSEMBLEUR - DÉSASSEMBLEUR DE PAQUETS: ADP, . . . . .	90
---	----

### ANNEXE J:

AMANDÉ RECOMMENDATIONS X.28 AND X.29, . . . . .	110
---	-----

## A N N E X E    A

### DESCRIPTION GÉNÉRALE ET RÈGLES DE FONCTIONNEMENT APPLICABLES

#### AUX DIAGRAMMES DES ÉTATS

- 1 - Chaque diagramme des états ne se trouve à un moment donné quelconque que dans un seul état.
- 2 - Chaque état est représenté par une ellipse qui comprend un numéro de référence et la désignation de l'état.
- 3 - Les passages admissibles d'un état à un autre sont représentés par des lignes de connexion avec une flèche signalant le sens permis du passage et indiquant l'événement ou les événements responsables de cette transition.
- 4 - Lorsqu'une transition résulte de tel ou tel état, elle peut être désignée par une flèche large aboutissant à l'état de description et indiquer les états d'origine admissibles, avec l'événement ou les événements qui en sont la cause.
- 5 - Un événement est soit l'émission (E) ou la réception (R) d'une commande ou d'une réponse ou une opération locale.
- 6 - Chaque diagramme des états comporte un état de "repos" portant le numéro zéro. C'est l'état initial ou de réinitialisation qui caractérise un état inactif.



- 7 - Lorsqu'on émet une commande exigeant une réponse, l'émission de commandes supplémentaires n'est pas permise aussi longtemps que cette réponse n'a pas été reçue. Un compteur de temporisation est déclenché et si aucune réponse n'est reçue avant expiration du délai prévu, l'émission d'une commande d'interruption de l'échange est obligatoire soit directement si une commande CSA a été émise, soit par l'envoi d'une CSA.
- 8 - L'effet produit par chaque événement responsable d'un passage d'état à un autre doit avoir cessé avant l'événement qui lui succède.
- 9 - Pendant un échange, chaque partenaire doit veiller à ce que l'opération se déroule correctement:
  - a - l'application du rôle de source/collecteur actuellement accepté;
  - b - utilisation appropriée des séries de procédures de commande/réponse, selon la description faite dans les diagrammes des états et dans les règles régissant leur fonctionnement;
  - c - surveillance pendant une période d'inactivité (par exemple, avec indication d'une défaillance ou d'un autre dérèglement empêchant une utilisation productive de l'échange).

Si on détecte une défaillance empêchant la poursuite normale de l'opération décrite plus haut, il faut obligatoirement recourir aux procédures de correction des erreurs définies pour chaque diagramme des états ou, si elles ne sont pas définies de manière précise, mettre fin à l'échange (interruption anormale).

Cette opération est indispensable afin d'éviter une utilisation improductive des services télétext pouvant entraîner des frais inutiles lorsqu'il n'est pas utilisé efficacement et donc causer une dégradation de la qualité du service.

- 10 - Les diagrammes des états doivent aider à définir l'utilisation appropriée des éléments de la procédure et non à décrire en détail une mise en oeuvre particulière.



## A N N E X E    B

### GRAPHES D'ÉTATS DU NIVEAU TRANSPORT

=====

Nous reprenons dans cette annexe les différents graphes du niveau transport. Ceux-ci ont déjà été évoqués au chapitre V paragraphe 1.5.

Voici la liste des différentes figures reprises dans cette annexe:

- B.1 - La liste des primitives et des paramètres fondamentaux tels qu'ils ont été retenus par le CCITT.
- B.2 - Le graphe correspondant à la phase d'initialisation.
- B.3 - Le graphe correspondant à la phase d'établissement et de libération: côté appelant.
- B.4 - Le graphe correspondant à la phase d'établissement et de libération: côté appelé.
- B.5 - Le graphe correspondant à la phase de transfert de données: émission.
- B.6 - Le graphe correspondant à la phase de transfert de données: réception.
- B.7 - La table séquentielle correspondant à la phase d'établissement et de libération: côté appelant.
- B.8 - La table séquentielle correspondant à la phase d'établissement et de libération: côté appelé.



B.9 - Les tables séquentielles correspondant à la phase de transfert de données: émission et réception.

B.10- Les graphes relatifs aux stations appelantes et appelées en ne retenant que les primitives propres.

B.11- Les tables séquentielles traduisant les graphes de l'annexe B.10.

## ANNEXE B1

## ECMA CLASSE/O (TELETEX)

	OCTET 1	OCTET 2	OCTET 3	OCTET 4	OCTET 5	OCTET 6	OCTET 7
TCR	Length	11100000	00000000	00000000	Source	reference	00000000 Parameters

TCA	Length	11010000	Destination	reference	Source	reference	00000000 Parameters
-----	--------	----------	-------------	-----------	--------	-----------	---------------------

TCC	Length	10000000	Destination	reference	Source	reference	Clearing cause Parameters
-----	--------	----------	-------------	-----------	--------	-----------	---------------------------

TCA

TBR	Length	01110000	Destination	reference	Reject cause	Parameters
-----	--------	----------	-------------	-----------	--------------	------------

TDT	Length	11110000	00000000	Data
-----	--------	----------	----------	------

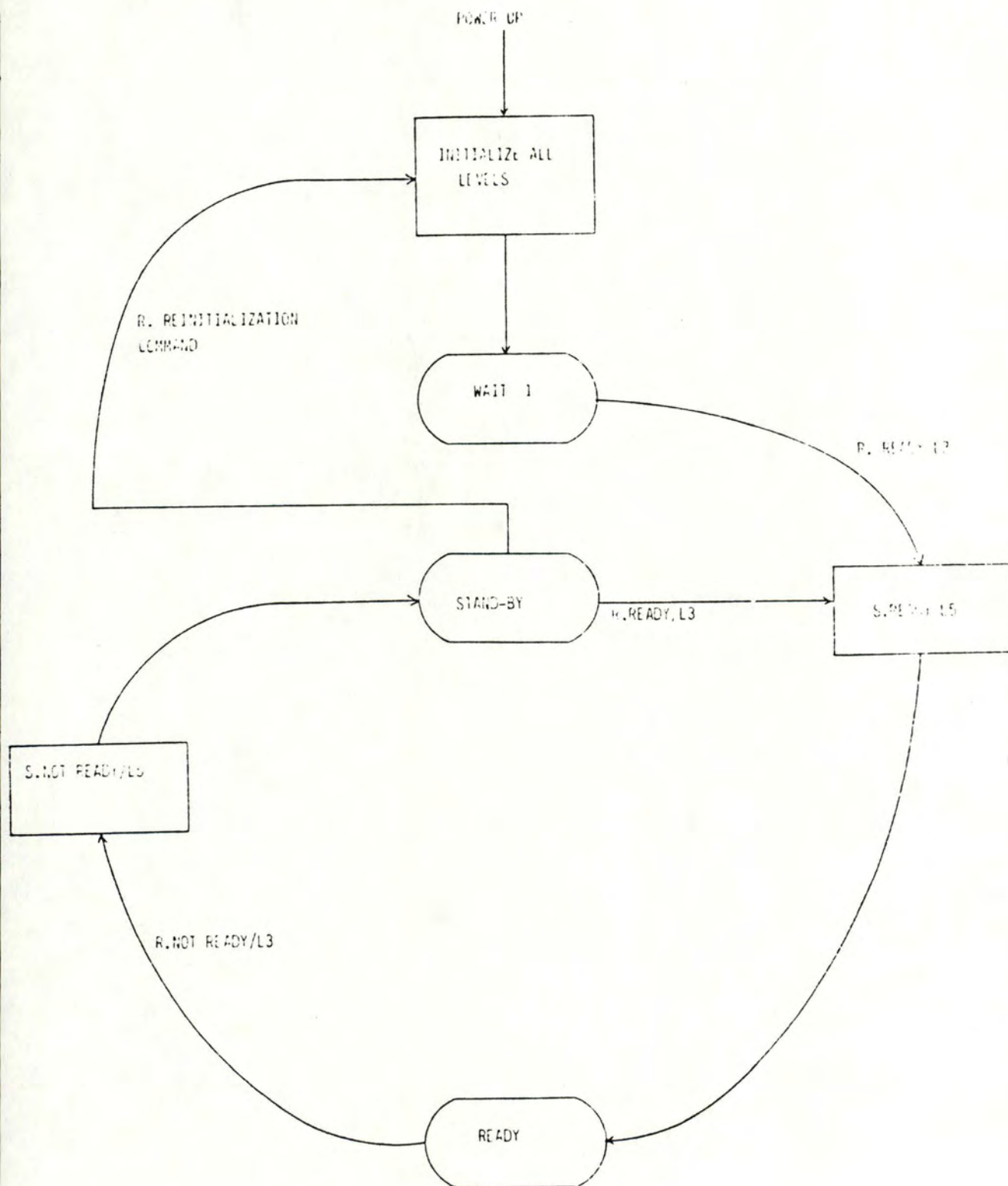
TSDU end mark

CCITT-43971

*Négociation* → *claire de T* *droit de fenêtre* *long* *TPP*

BLOCS UTILISES POUR LE PROTOCOLE DU NIVEAU TRANSPORT

## ANNEXE B2



INITIALIZATION  
GLOBAL FOR THE TERMINAL



## ANNEXE B3

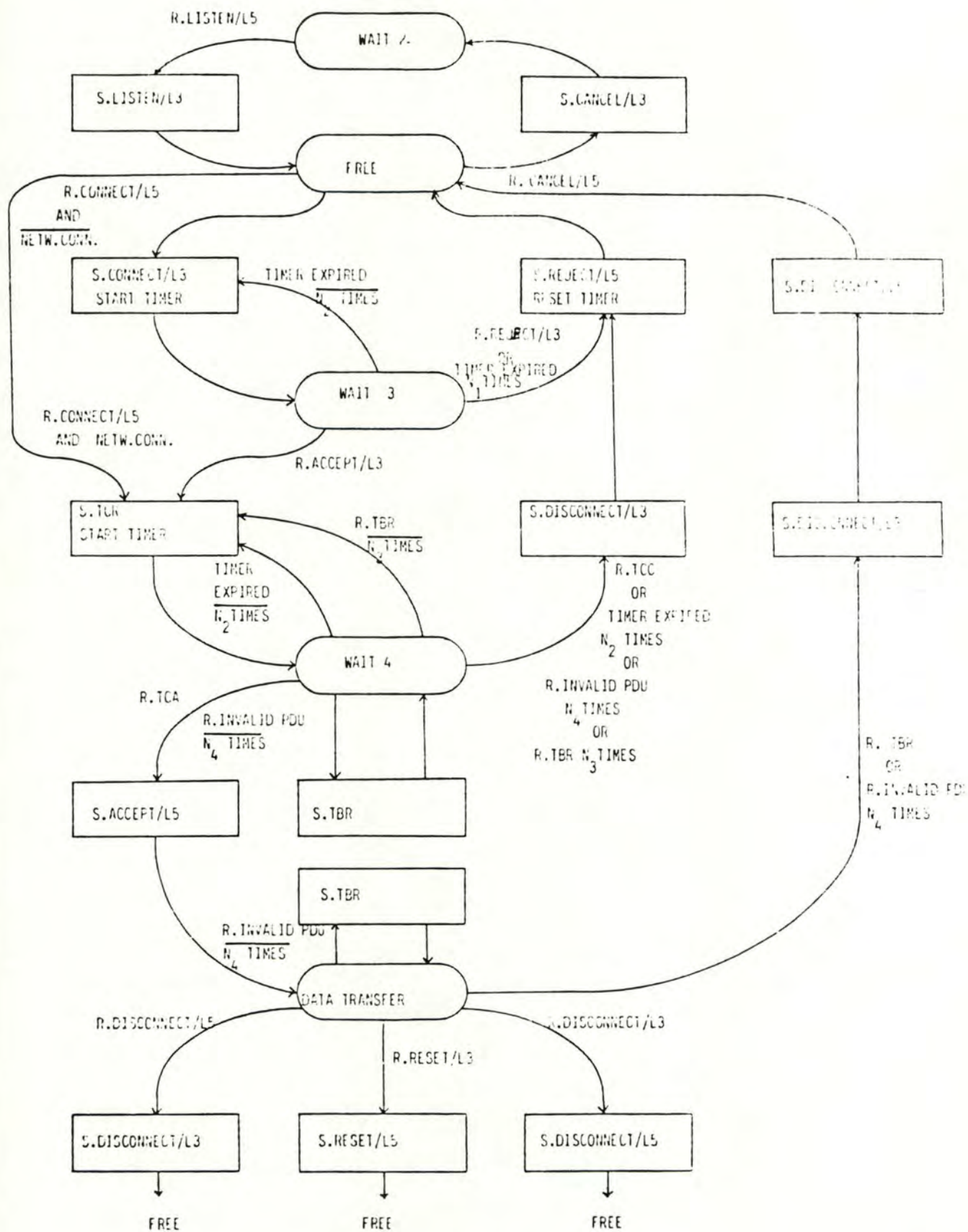
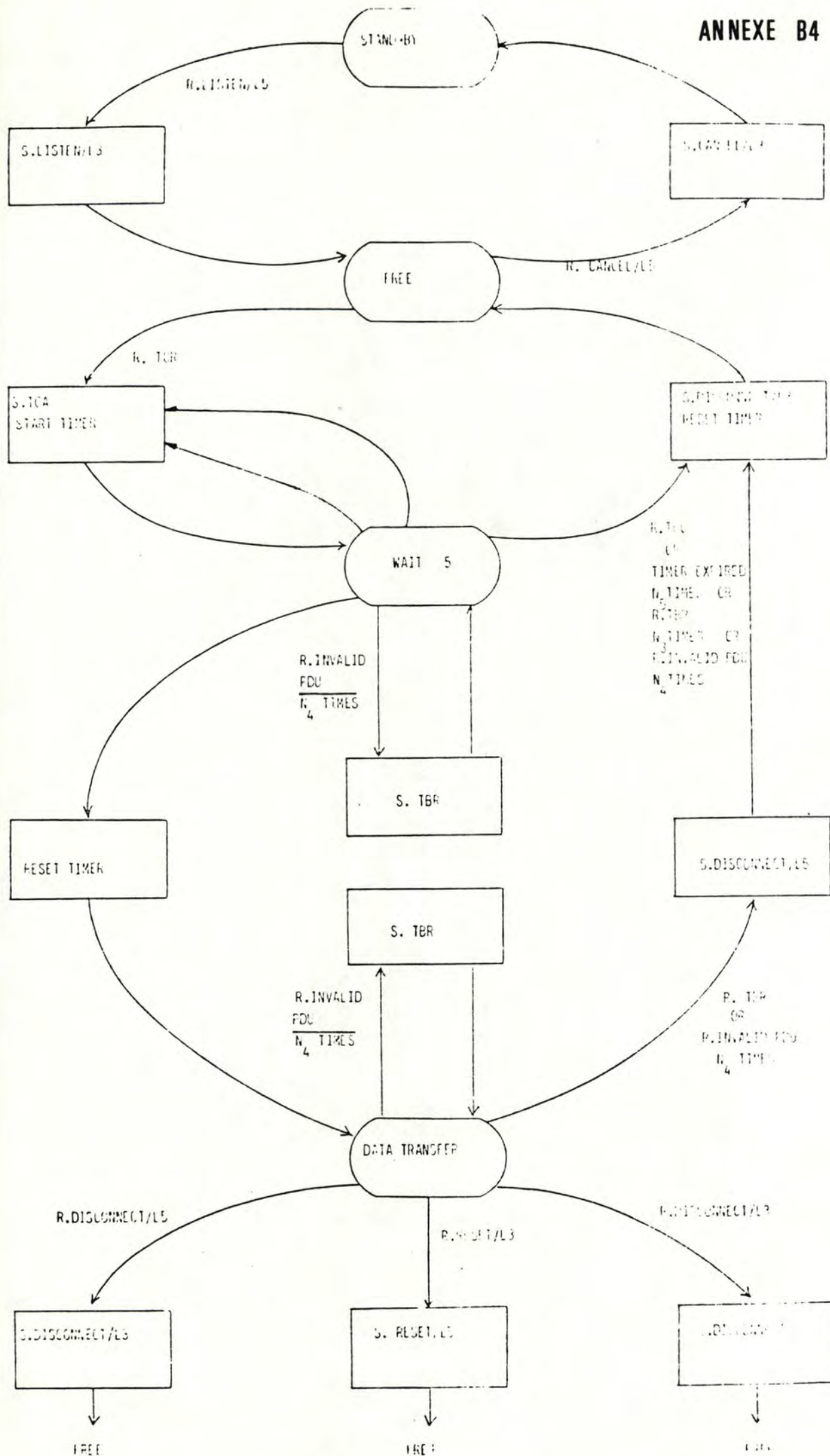


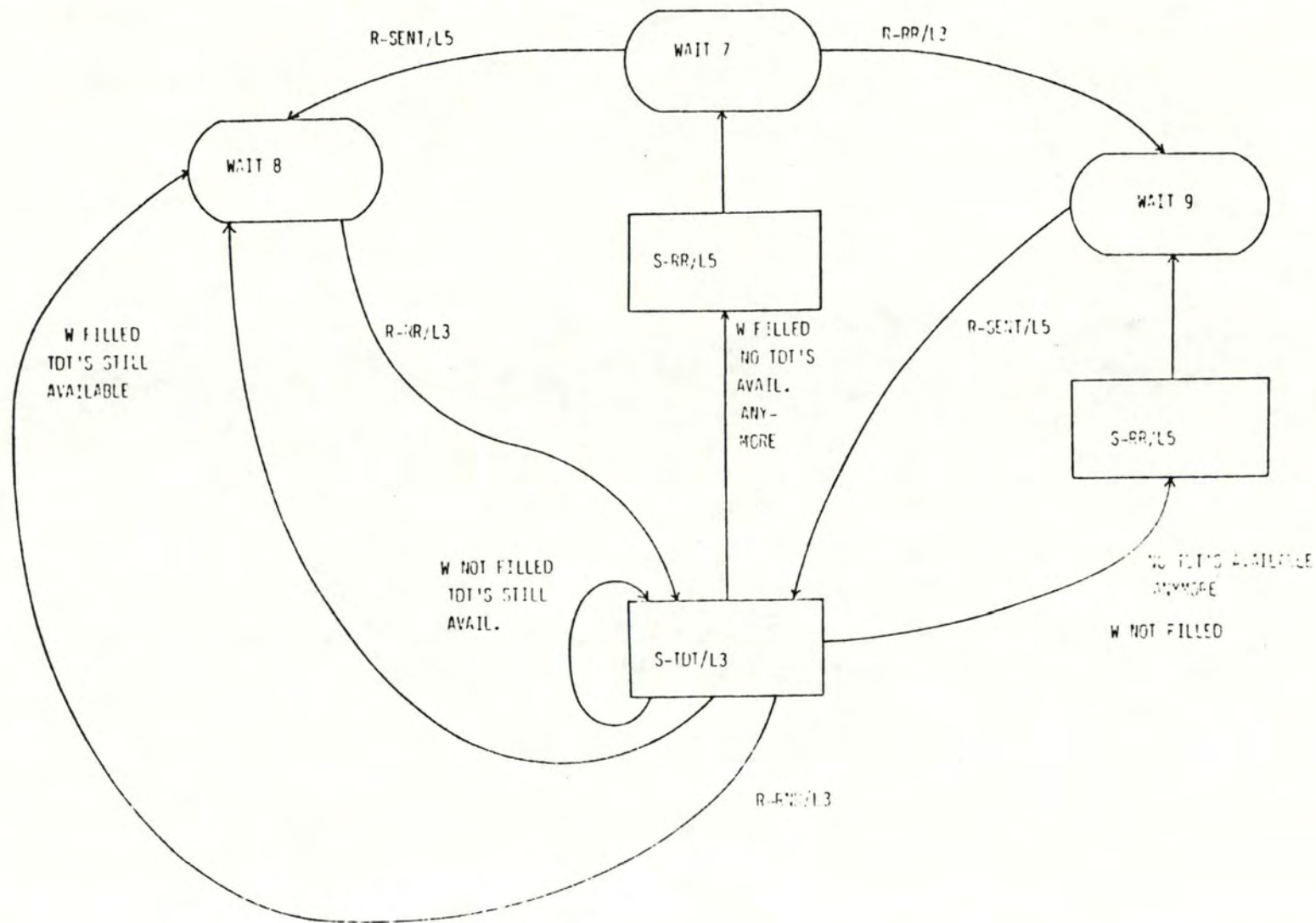
FIG 2/3

N1 To N4 = 2

TIMERS = 5 sec.

ASSOCIATION ESTABLISHMENT AND TERMINATION : CALLING SIDE  
PER LOGICAL CHANNEL



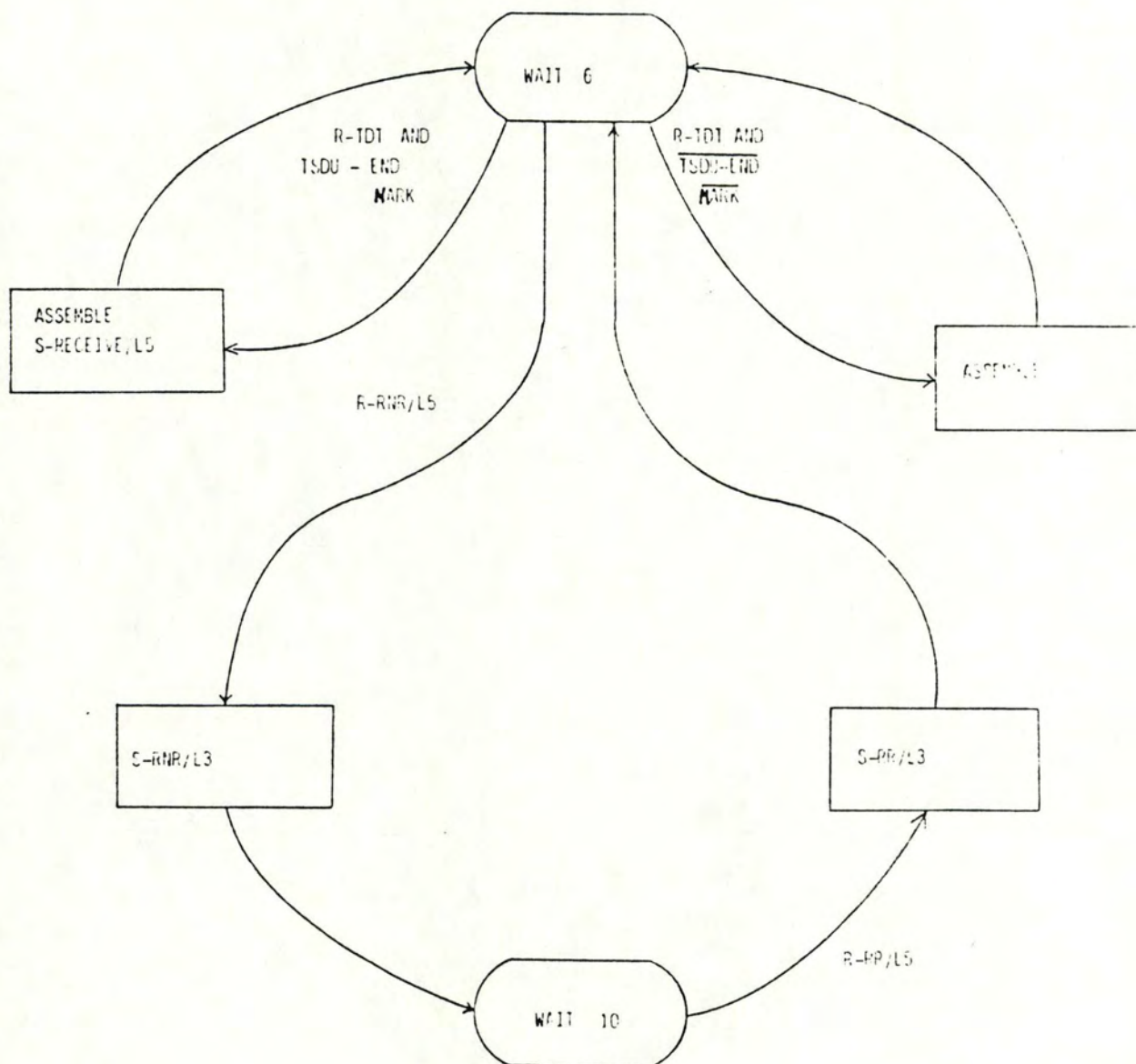


ANNEXE B5



## ANNEXE B6

DATA TRANSFER : RECEPTION.



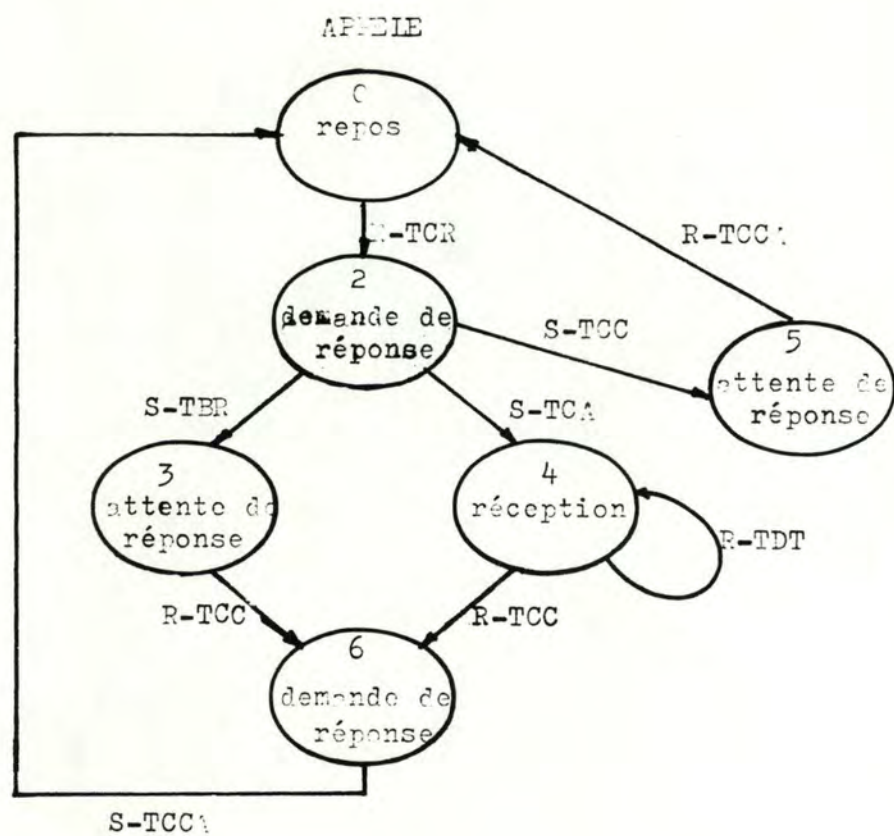
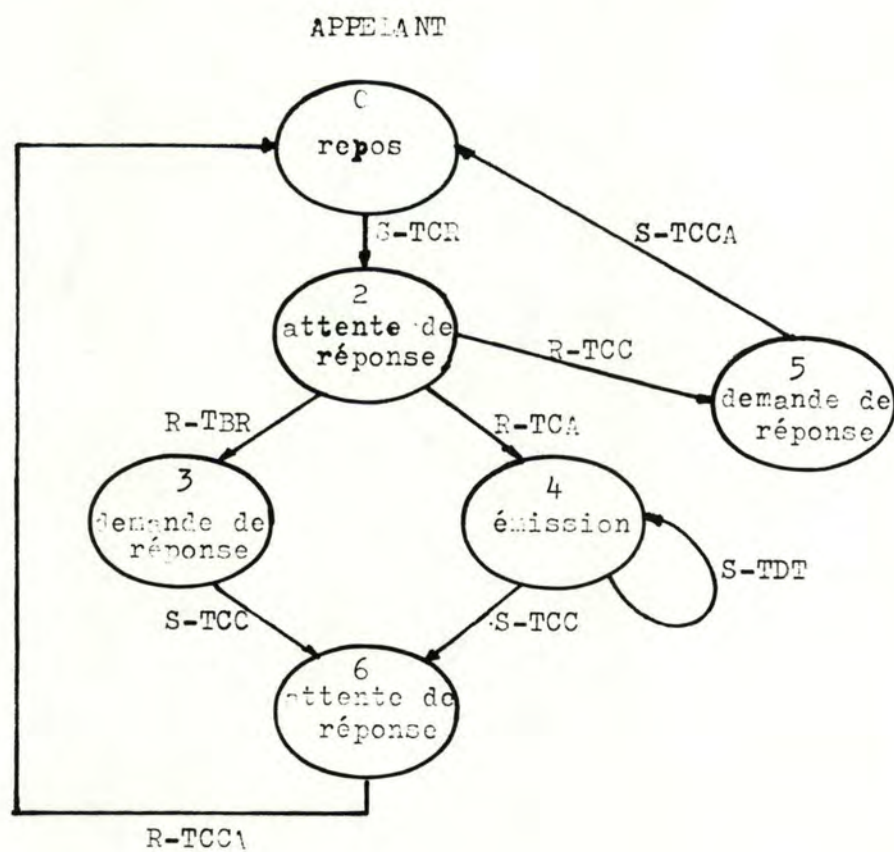








## ANNEXE B 10



## ANNEXE B11

## APPELANT

Dénomination	Ph	reçoit				envoie			
		T	T	T	T	T	T	T	T
		B	C	C	C	C	C	D	C
		R	A	C	A	R	C	T	A
Repos	1					2			
Attente de réponse	2	3	4	5					
Demande de réponse	3					6			
Emission	4					6	4		
Demande de réponse	5							1	
Attente de réponse	6				1				

## APPELÉ

Dénomination	Ph	reçoit				envoie			
		T	T	T	T	T	T	T	T
		C	C	D	C	B	C	C	C
		R	C	T	A	R	A	C	A
Repos	1	2							
Demande de réponse	2					3	4	5	
Attente de réponse	3	6							
Réception	4	6	4						
Attente de réponse	5				1				
Demande de réponse	6							1	



## A N N E X E    C

### GRAPHES D'ÉTATS DU NIVEAU SESSION

=====

Nous reprenons dans cette annexe les différents graphes du niveau session. Ceux-ci ont déjà été évoqués au chapitre V.

Voici la liste des différentes figures reprises dans cette annexe:

- C.1 - La liste des primitives du niveau session.
- C.2 - La liste des paramètres associés aux primitives de l'annexe C 1.
- C.3 - Le graphe correspondant au diagramme d'états et de transitions: côté appelant.
- C.4 - Le graphe correspondant au diagramme d'états et de transitions: côté appelé.
- C.5 - Les tables séquentielles correspondant aux annexes C.3 et C.4.
- C.6 - L'organigramme correspondant au diagramme d'états: côté appelant.
- C.7 - L'organigramme correspondant au diagramme d'états: côté appelé.
- C.8 - Un tableau comparatif entre les primitives de niveau session présentées par le CCITT, l'ECMA, le conversationnel.

Le lecteur intéressé par une description détaillée de ces primitives et de leurs paramètres pourra trouver des renseignements supplémentaires dans le "Projet Avis S-62" du CCITT.

## ANNEXE C1

## N 5 - SESSION COMMANDS AND RESPONSES

COMMAND	RESPONSE	ABBREVIATION
SESSION ESTABLISHMENT AND CLEARING		
COMMAND SESSION START	RESPONSE SESSION START POSITIVE	CSS RSSP
	RESPONSE SESSION START NEGATIVE	RSSN
COMMAND SESSION END	RESPONSE SESSION END POSITIVE	CSE RSEP
COMMAND SESSION ABORT	RESPONSE SESSION ABORT POSITIVE	CSA RSAP
INFORMATION TRANSFER		
COMMAND SESSION USER INFORMATION	RESPONSE SESSION USER INFORMATION	CSUI RSUI
SESSION MANAGEMENT		
COMMAND SESSION CHANGE CONTROL	RESPONSE SESSION CHANGE CONTROL POSITIVE	CSCC RSCCP

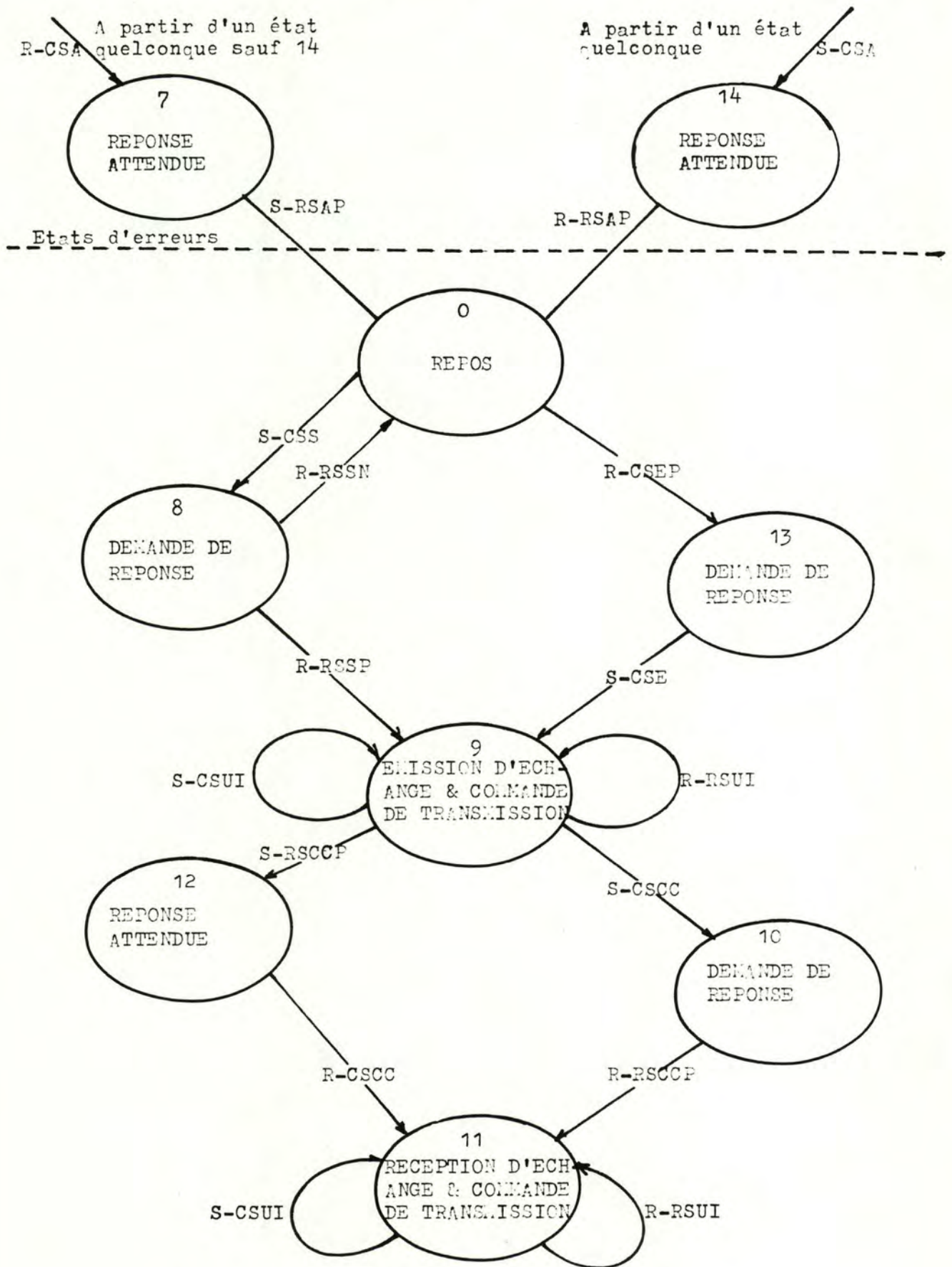


## PGIs AND PLs FOR SESSION ELEMENTS OF PROCEDURE

SESSION COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY
CSS	SESSION REFERENCE	M	TERMINAL IDENTIFICA- TION OF THE CALLED TERMINAL	M
			DATE AND TIME	M
			ADDITIONAL SESSION REFERENCE NUMBER	NM
	NONBASIC SESSION CAPABILITY	NM	WINDOW SIZE	NM
			MISCELLANEOUS SESSION CAPABILITIES	NM
	NONBASIC TERMINAL CAPABILITIES	NM	CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANEOUS TERMI- NAL CAPABILITIES	NM
	PRIVATE USE	NM	—	—
			SERVICE IDENTIFIER	M
CSE			SESSION TERMINATION	NM
CSA			SESSION TERMINATION	M
CSCC				
CSUI				



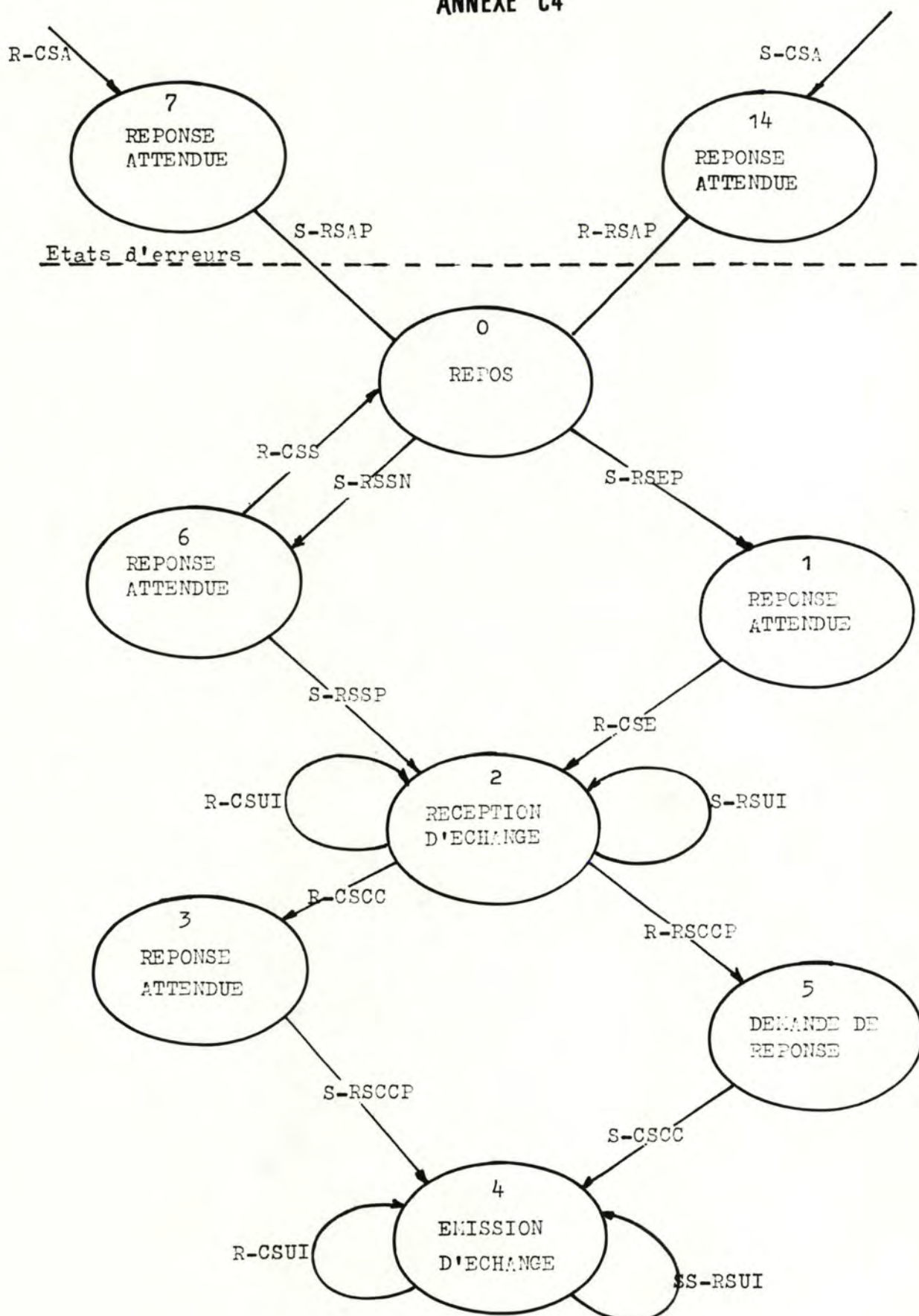
SESSION COMMAND OR RESPONSE IDENTIFIER	PARAMETER GROUP IDENTIFIERS (PGI)		PARAMETER IDENTIFIERS (PI)	
	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY	DESCRIPTION	MANDA- TORY OR NOT MAN- DATORY
RSSP	SESSION REFERENCE	M	TERMINAL IDENTIFICA- TION OF THE CALLED TERMINAL	M
			DATE AND TIME	M
			ADDITIONAL SESSION REFERENCE NUMBER	NM
	NONBASIC SESSION CAPABILITIES	NM	WINDOW SIZE	NM
			MISCELLANCOUS SES- SION CAPABILITIES	NM
	NONBASIC TERMINAL CAPABILITIES	NM	GRAPHIC CHARACTER SETS	NM
			CONTROL CHARACTER SETS	NM
			PAGE FORMATS	NM
			MISCELLANCOUS TER- MINAL CAPABILITIES	NM
			SERVICE IDENTIFIER	M
	PRIVATE USE	NM	—	—
			REQUEST SESSION FUNCTIONS	NM
RSSN	PRIVATE USE			NM
	FOR FURTHER STUDY	--	—	—
RSEP				
RSAP				
RSCCP				
RSUI			REQUEST SESSION FUNCTIONS	NM



TERMINAL APPELANT.



## ANNEXE C4



TERMINAL APPELE.



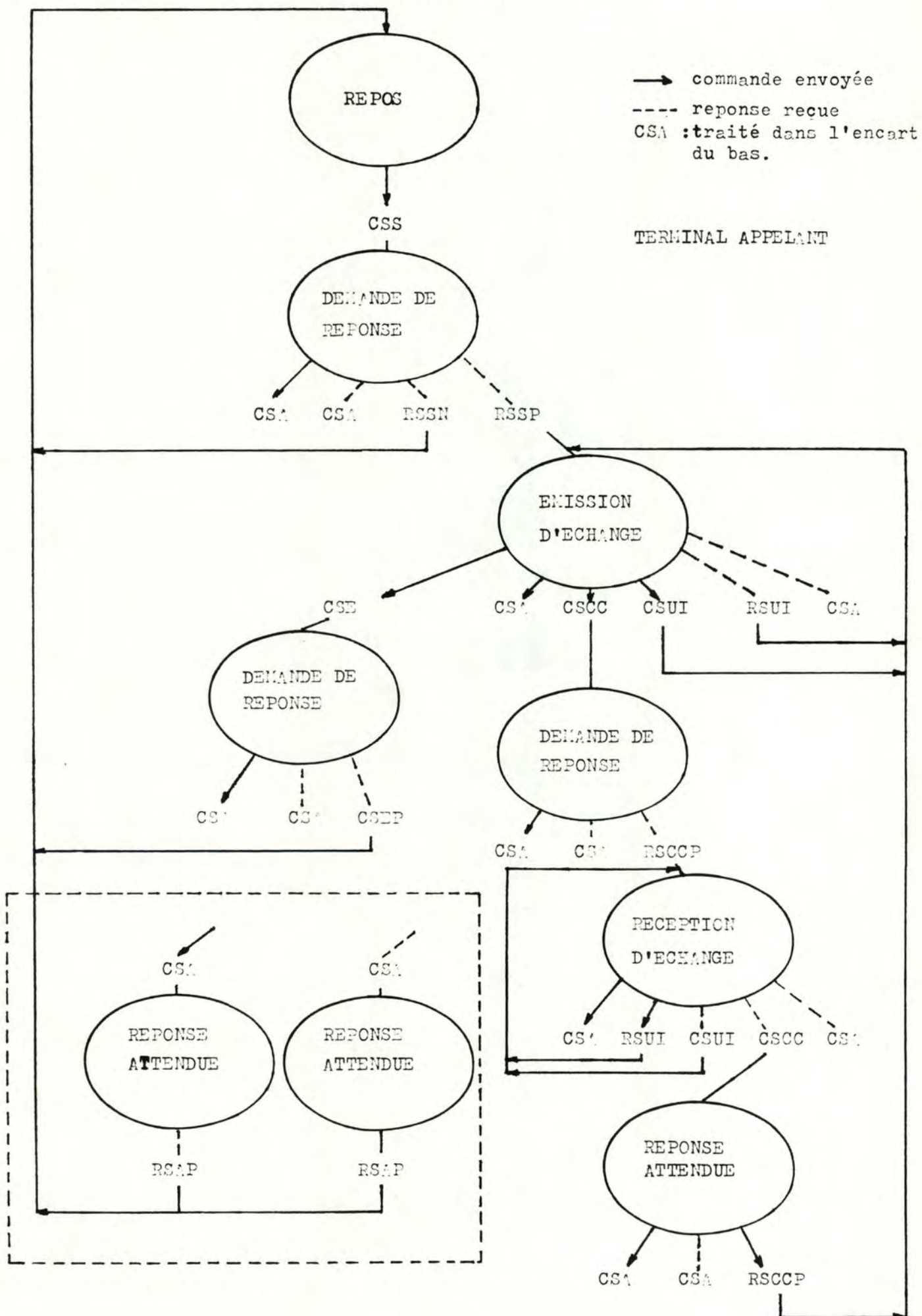
## Reçu                      Envoie

APPELÉ

Envoie

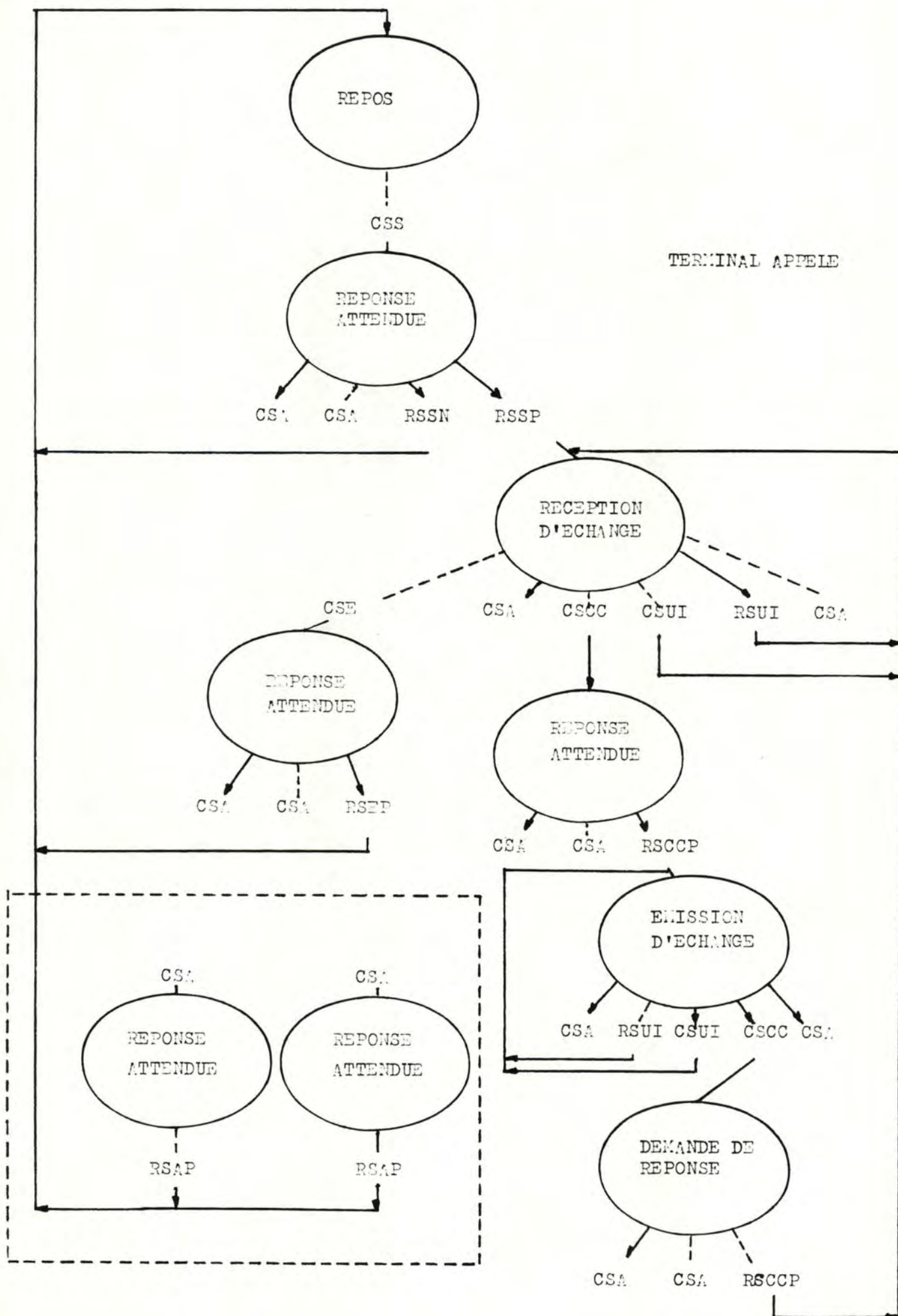
[illegible]

## ANNEXE C6





## ANNEXE C7





## ANNEXE C8

## P R I M I T I V E S D E S E S S I O N

<u>C.C.I.T.T.</u>	<u>MEYER (ECMA)</u>	<u>CONVERSATIONNEL</u>
1. CSS , RSSP RSSN	S. CONNECT, R. CON	S. CONNECT
2. CSE , RSEP	S. RELEASE, R. NOTFINISHED R. FINISHED	S. RELEASE
3. CSA , RSAP	S. ABORT, R. ABORT	S. ABORT
4. CSUI , (RSUI)	S. DATA, R. DATA S SEND S. DATE-REC, R. DATA. R	S. DATA
5. CCCC , RSCCP	S. ACTIVATE, R. ACTIVATE	
S. EXPED, R. EXPED	S. EXPED, R. EXPED	S. EXPED
S. TOKEN GIVE, R. TOK G.	S. TOKEN GIVE, R. TOK G.	S. TOKEN G
S. TOKEN PLEASE, R. TOK	S. TOKEN PLEASE, R. TOK P.	S. TOKEN P.
S. TOKEN RELEASE	S. TOKEN RELEASE	S. TOKEN R.
S. NUT FINISHED	S. NUT FINISHED	
	S. FINISHED	S. RESECT en complé- ment de S. CONNECT
		S. SUSPEND
		S. REACTIVATE en complément S. SUSPEND
		S. QUARANTINE SERVICE
		S. QUARANTINE CANCEL

## A N N E X E    D

### FONCTION DE PRÉSENTATION

#### DU CODE ASCII ET ASCII ÉTENDU

=====

On trouve dans cette annexe l'illustration des différentes fonctions des codes ASCII. Leur représentation a déjà été évoquée au chapitre V.

Voici la liste des différentes descriptions des codes ASCII:

- D.1 - Code ASCII à 5 moments.
- D.2 - Signification sommaire des caractères de commande.
- D.3 - Caractéristiques fonctionnelles des caractères de commande au niveau présentation (Code ASCII).
- D.4 - Caractéristiques fonctionnelles des caractères de commande au niveau présentation (Code ASCII étendu).
  - + A - Répertoire des fonctions de commande.
  - + B - Caractères propres à l'ASCII étendu.
  - + C - Les sous-répertoires.
  - + D - Fonctions de commande.
  - + E - Exemple: fonction de présentation.
- D.5 - Système d'identification des commandes pour l'ASCII étendu.

Le lecteur intéressé par ces codes ASCII trouvera de plus amples explications dans les publications du CCITT.

Tome VIII.1 - Avis V3

Tome VIII.1 - Avis V4.



## ANNEXE D1

ALPHABET INTERNATIONAL. N° 5

					b <sub>7</sub>	0	0	0	0	1	1	1	1
					b <sub>6</sub>	0	0	1	1	0	0	1	1
					b <sub>5</sub>	0	1	0	1	0	1	0	1
						0	1	2	3	4	5	6	7
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>		0	0	0	0	0	1	1	1	1
0	0	0	0	0	NUL	TC <sub>7</sub> (DLE)	SP	0	@	P	'	p	
0	0	0	1	1	TC <sub>1</sub> (SOH)	DC <sub>1</sub>	!	1	A	Q	a	q	
0	0	1	0	2	TC <sub>2</sub> (STX)	DC <sub>2</sub>	"	2	B	R	b	r	
0	0	1	1	3	TC <sub>3</sub> (ETX)	DC <sub>3</sub>	#	3	C	S	c	s	
0	1	0	0	4	TC <sub>4</sub> (EOT)	DC <sub>4</sub>	□	4	D	T	d	t	
0	1	0	1	5	TC <sub>5</sub> (ENQ)	TC <sub>8</sub> (NAK)	%	5	E	U	e	u	
0	1	1	0	6	TC <sub>6</sub> (ACK)	TC <sub>9</sub> (SYN)	&	6	F	V	f	v	
0	1	1	1	7	BEL	TC <sub>10</sub> (ETB)	'	7	G	W	g	w	
1	0	0	0	8	FE <sub>0</sub> (BS)	CAN	(	8	H	X	h	x	
1	0	0	1	9	FE <sub>1</sub> (HT)	EM	)	9	I	Y	i	y	
1	0	1	0	10	FE <sub>2</sub> (LF)	SUB	*	:	J	Z	j	z	
1	0	1	1	11	FE <sub>3</sub> (VT)	ESC	+	;	K	[	k	{	
1	1	0	0	12	FE <sub>4</sub> (FF)	IS <sub>4</sub> (FS)	,	<	L	\	l	†	
1	1	0	1	13	FE <sub>5</sub> (CR)	IS <sub>3</sub> (GS)	-	=	M	]	m	}	
1	1	1	0	14	SO	IS <sub>2</sub> (RS)	.	>	N	^	n	-	
1	1	1	1	15	SI	IS <sub>1</sub> (US)	/	?	O	_	o	DEL	



## ANNEXE D2

ALPHABET INTERNATIONAL N° 5CARACTERES DE COMMANDEAbréviationsSignification

ACK	Accusé de réception
<u>BEL</u>	<u>Sonnerie</u>
<u>BS</u>	<u>Retour arrière</u>
CAN	Annulation
<u>CR</u>	<u>Retour du chariot</u>
DC	Commande d'appareil auxiliaire
DEL	Oblitération
<u>DLE</u>	<u>Echappement transmission</u>
EM	Fin de support
ENQ	Demande
EOT	Fin de communication
<u>ESC</u>	<u>Echappement</u>
ETB	Fin de bloc de transmission
<u>ETX</u>	<u>Fin de texte</u>
FE	Commande de mise en page
<u>FF</u>	<u>Page suivante</u>
FS	Séparateur de fichier
<u>GS</u>	<u>Séparateur de groupe</u>
<u>HT</u>	<u>Tabulation horizontale</u>
<u>IS</u>	<u>Séparateur d'information</u>
<u>LF</u>	<u>Interligne</u>
NAK	Accusé de réception négatif
<u>NUL</u>	<u>Nul</u>

<u>RS</u>	<u>Séparateur d'article</u>
<u>SI</u>	<u>En code</u>
<u>SO</u>	<u>Hors code</u>
SOH	Début d'en-tête
<u>SP</u>	<u>Espace</u>
<u>STX</u>	<u>Début de texte</u>
SUB	Substitution
SYN	Synchronisation
TC	Commande de transmission
<u>US</u>	<u>Séparateur de sous-article</u>
<u>VT</u>	<u>Tabulation verticale</u>

## ANNEXE D3

### CARACTÉRISTIQUES FONCTIONNELLES DES CARACTÈRES DE COMMANDE AU NIVEAU PRÉSENTATION (CODE ASCII).

#### 1 - CARACTÈRES DE MISE EN PAGE (F.E)

sur imprimante ou récepteur visuel.

Les définitions de commande de mise en page emploient les concepts suivants:

- a - une page est composée d'un nombre défini de lignes de caractères;
- b - les caractères formant une ligne occupent un nombre défini de positions appelées positions de caractère;
- c - la position active est la position de caractère dans laquelle le caractère, sur le point d'être traité, apparaîtrait s'il était à imprimer. Normalement, la position active se déplace d'une position de caractère à la fois.

BS Retour arrière

Ramène la position active en arrière d'une position de caractère sur la même ligne.

HT Tabulation horizontale

Déplace la position active jusqu'à la position de caractère prédéterminée suivante sur la même ligne.

LF Interligne

Caractère correspondant sur la ligne suivante.

VT Tabulation verticale

Déplace la position active jusqu'à la position de caractère correspondante sur la ligne suivante prédéterminée.



FF Page suivante

Commande de mise en page qui déplace la position active jusqu'à la position de caractère correspondante sur une ligne prédéterminée d'un imprimé ou d'une page suivante.

CR Retour chariot

Position active devient la première position de la ligne suivante.

2 - COMMANDES D'ORGANE PERIPHERIQUE (D C)

Caractère de commande destiné à la commande d'un ou plusieurs organes périphériques situés sur place ou éloignés et reliés à un système de traitement des données ou de télécommunication.

Ces caractères ne sont pas prévus pour commander des systèmes de télécommunication; ceci doit se faire par l'intermédiaire des T C (voir annexe D.2)

Commandes d'appareil auxiliaire

- DC1 - Caractère de commande d'appareil auxiliaire principalement destiné à enclencher ou à mettre en marche un appareil auxiliaire. Si on n'en a pas besoin pour cette fonction, il peut être utilisé pour rétablir dans un appareil le mode principal de fonctionnement (voir aussi DC2 et DC3) ou pour toute autre fonction de commande d'appareil auxiliaire non prévue par les autres DC.
- DC2 - Caractère de commande d'appareil auxiliaire principalement destiné à enclencher ou à mettre en marche un appareil auxiliaire. Si on n'en a pas besoin pour cette fonction, il peut être utilisé afin que l'appareil fonctionne d'après un mode spécial (dans ce cas, DC1 sera utilisé pour ramener

l'appareil au mode principal de fonctionnement)  
ou pour toute autre fonction de commande d'appareil auxiliaire non prévue par les autres DC.

DC3 - Caractère de commande d'appareil auxiliaire principalement destiné à déclencher ou à arrêter l'appareil auxiliaire. Cette fonction peut être un arrêt de niveau secondaire, par exemple attente, pause, mise en réserve ou halte (dans ce cas, DC1 est utilisé pour rétablir l'opération normale) Si on n'en a pas besoin pour cette fonction, il peut être utilisé pour toute autre fonction de commande d'appareil non prévue par les autres DC.

DC4 - Caractère de commande d'appareil auxiliaire principalement destiné à déclencher, arrêter ou interrompre un appareil auxiliaire. Si on n'en a pas besoin pour cette fonction, il peut être utilisé pour toute autre fonction de commande d'appareil auxiliaire non prévue par les autres DC.

Exemples d'usage de commandes d'appareil auxiliaire:

1 - Une connexion

marche - DC2    arrêt - DC4

2 - Deux connexions indépendantes

Première connexion marche - DC2    arrêt - DC4

Seconde connexion marche - DC1    arrêt - DC3

3 - Deux connexions dépendantes

Général                    marche - DC2    arrêt - DC4

Particulier                marche - DC1    arrêt - DC3

4 - Connexion de l'entrée et de la sortie

Sortie                    marche - DC2    arrêt - DC4

Entrée                    marche - DC1    arrêt - DC3



### 3 - SEPARATEURS D'INFORMATIONS (I S)

utilisés pour séparer et qualifier logiquement des données. Ils peuvent être utilisés dans un ordre hiérarchique supérieur ou non hiérarchique. Dans le second cas, leur signification spécifique dépend de leur application.

S'ils sont utilisés hiérarchiquement, l'ordre décroissant est:

- FS Caractère de commande employé pour séparer et qualifier des données dans un sens logique; sa signification spécifique doit être déterminée pour chaque application. Si ce caractère est employé dans l'ordre hiérarchique il délimite un ensemble de données appelé "FICHIER".
- GS Si ce caractère est employé dans un ordre hiérarchique indiqué dans la définition générale de IS, il détermine un ensemble de données appelé "GROUPE".
- RS Il détermine un ensemble de données appelé "ARTICLE".
- US ... Il délimite un ensemble de données appelé "SOUS-ARTICLE".

### 4 - CARACTERES PARTICULIERS

NUL Nul

destiné au remplissage de temps ou de support d'information.

- Des caractères NUL peuvent être ajoutés ou retirés sans que le contenu d'information de cette suite en soit affecté, seulement dans ce cas, l'adjonction ou la suppression de ces caractères peut modifier la disposition des informations et/ou la commande des équipements.



- BEL** Sonnerie.  
Utilisé lorsqu'il est nécessaire d'attirer l'attention  
→ peut commander des dispositifs d'appels.
- S0** Hors code.  
Caractère de commande qui est employé en combinaison avec les caractères EN CODE et ECHAPPEMENT pour étendre le jeu de caractère graphique du code.  
Rem. SP et DEL non affectés.
- SI** En code.  
Caractère de commande qui est employé en combinaison avec les caractères HORS-CODE et ECHAPPEMENT pour étendre le jeu de caractères graphique du code.
- CAN** Annulation.  
Caractère (ou les caractères) d'une suite indiquant que les données le précédant sont erronées et doivent être ignorées.
- EM** Fin de support.  
Permet d'identifier la fin matérielle du support ou la fin de la partie utilisée du support ou la fin de la partie désirée des informations enregistrées sur un support.
- SUB** Caractère de substitution.  
Employé pour remplacer un caractère reconnu non valide ou erroné.  
SUB est introduit par le système de traitement.
- ESC** Echappement.  
Il modifie la signification d'un nombre limité de combinaisons d'éléments successifs qui le suivent et constituent la séquence d'échappement.  
(ex. fournir des jeux de caractères graphiques).  
- Ces commandes supplémentaires ne doivent pas être utilisées comme commandes additionnelles de transmission.

SP Espace.

Déplace la position active d'un caractère sur la même ligne (  $\emptyset$  )

DEL Oblitération

- effacer/oblitérer des caractères erronés ou indésirables sur une bande perforée.
- servir de caractères de remplissage de temps ou de support d'information.
- Ils peuvent être insérés ou retirés d'une suite de caractères sans que le contenu de l'information soit affecté. Dans ce cas, l'insertion ou la suppression de ces caractères peut affecter la disposition des informations de la commande des équipements.

## ANNEXE D4

CARACTÉRISTIQUES FONCTIONNELLES DES CARACTÈRES DE COMMANDE

AU NIVEAU PRÉSENTATION (CODE ASCII ETENDU)



ASCII ETENDU

					<table><tr><td>c</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>b</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>b</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>b</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>																c	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	b	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	b	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	b	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
c	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1																																																																								
b	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1																																																																								
b	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1																																																																								
b	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1																																																																								
					<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr></table>																0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																																																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																																																																									
0	0	0	0	0			SP	0	@	P		p			°			Ω	k																																																																					
0	0	0	1	1			!	1	A	Q	a	q			i	±	`	Æ	œ																																																																					
0	0	1	0	2			"	2	B	R	b	r			e	²	/	ð	đ																																																																					
0	0	1	1	3				3	C	S	c	s			f	³	^	ä	ð																																																																					
0	1	0	0	4				4	D	T	d	t			\$	x	~	H	h																																																																					
0	1	0	1	5			%	5	E	U	e	u			¥	μ	-		ı																																																																					
0	1	1	0	6			&	6	F	V	f	v			#	¶	˘	IJ	ij																																																																					
0	1	1	1	7			'	7	G	W	g	w			§	•	•	Ł	ł																																																																					
1	0	0	0	8	BS		(	8	H	X	h	x			¤	÷	¨	Ł	ł																																																																					
1	0	0	1	9			)	9	I	Y	i	y				°		Œ	œ																																																																					
1	0	1	0	10	LF	SUE	*	:	J	'Z	j	z				°		Œ	œ																																																																					
1	0	1	1	11		ESC	+	;	K	[	k		PLD	CSI	<<	>>	,	◊	β																																																																					
1	1	0	0	12	FF		,	<	L		L	I	PLU			¼	◊	Ł	ł																																																																					
1	1	0	1	13	CR		-	=	M	]	m					½	"	Ŧ	ẗ																																																																					
1	1	1	0	14			.	>	N		n					¾	˘	ŋ	ŋ																																																																					
1	1	1	1	15			/	?	O	◊	o					¿	˘	'n																																																																						

## A - RÉPERTOIRE DES FONCTIONS DE COMMANDE

- 1 - Caractères de mise en page.
- 2 - Fonction de commande de présentation.
- 3 - Fonction de commande d'extension de code.
- 4 - Fonction de commande diverses.

### 1 - CARACTERES DE MISE EN PAGE

<u>Représent.</u> <u>codée</u>	<u>Identificateur</u>	<u>Abréviation</u>	<u>Explication</u>
	SP01	SP	Espace (voir alphabet ascii caractère particulier)
0/8	CF10	BS	Retour arrière (voir ascii caractère de mise en page)
0/10	CF12	LF	Interligne
0/12	CF14	FF	Page suivante
0/13	CF15	CR	Retour du chariot
8/11	CF16	PLD	Interligne partiel vers le bas
8/12	CF17	PLU	Interligne partiel vers le haut

### 2 - COMMANDES POUR LA PRESENTATION

agissent sur :

- présentation de la page ( $\longleftrightarrow$ ,  $\updownarrow$ )
- espacement vertical
- mise en valeur graphique.

Remarque: les fonctions sont paramétrées.

Toutes les caractéristiques de présentation doivent être définies en début de page. L'espacement vertical et la mise en val. graph. peuvent être modifiée en cours de page.



CSI P <sub>1</sub>	Pm 2/0 4/10	CP01	PFS	Choix de la présentation de la page
CSI P <sub>1</sub>	Pm 6/13	CP03	SGR	Choix de la mise en valeur graphique
CSI P <sub>1</sub>	Pm 2/0 4/11	CP04	SHS	Choix de l'espacement horizontal
CSI P <sub>1</sub>	Pm 2/0 4/12	CP05	SVS	l'espacement vertical

### 3 - COMMANDE D'EXTENSION DE CODE

1/11	CE03	ESC	Echappement (voir alphabet ascii caractère particulier)
9/11	CE06	CSI	Introduceur de séquence de commande

### 4 - COMMANDES DIVERSES

1/10	CM02	SUB	Substitution (voir alphabet ascii caractère particulier).
CSI P <sub>1</sub> Pm2/0 4/13	CM04	IG5	Identification du sous-répertoire de caractères graphiques



## B - CARACTÈRES PROPRES À L'ASCII ÉTENDU

## 1 - CARACTERES DE MISE EN PAGE

CF16      PLD      Interligne partiel vers le bas (début de  
                              texte en inférieur / fin de texte en  
                              supérieur).

Fonction: amener la position active à la position de caractère correspondante sur une ligne "imaginaire" décalée verticalement d'un interligne partiel.

Les caractères qui suivent sont reproduits au-dessous de la ligne jusqu'à l'apparition de PLU, dans la suite de données.

CF17	PLU	Interligne partiel vers le haut (début de texte en supérieur / fin de texte en infé- rieur).
------	-----	--

Fonction: amener la position active à une position correspondante sur une ligne "imaginaire" décalée verticalement d'un inter-ligne partiel.

Les caractères qui suivent sont reproduits au-dessus de la ligne jusqu'à la première apparition d'un (PLD)

L'apparition d'un PLD ou PLU ne peut pas être suivie dans la même ligne d'un autre PLD ou PLU, mais bien d'un PLU ou PLD.

## 2 - CARACTERE POUR LA PRESENTATION

CP01	PLS	Choix de la présentation de la page: Commande de présentation avec paramètre sélectif qui définit la direction de la page.
------	-----	---

Les valeurs du paramètre ont le sens suivant:

```
o: format de base en hauteur
l: - - largeur
```

- CP03      SGR      Choix de la mise en valeur graphique:  
 Commande avec paramètre sélectif qui  
 définit une caractéristique de présen-  
 tation pour la suite du texte.  
 0: condition par défaut  
 4: souligné
- CP04      SHS      Choix de l'espacement horizontal:  
 Définit l'espacement des caractères pour  
 la suite du texte.  
 Peut être changée en tout point d'une  
 page et prend effet après le premier CR.LF  
 0: 10 caractères/ 25,4mm
- CP05      SVS      Choix de l'espacement vertical:  
 Définit l'espacement des lignes pour la  
 suite du texte.  
 Prend effet à l'apparition de la fonction  
 'Ligne suivante ou précédente'.  
 0: 6 lignes par 25,4 mm  
 1: 4 -      "  
 2: 3 -      "  
 3: 12 -      "      (sert pour le demi-interligne  
 vers le haut ou vers le bas).

### 3 - COMMANDE D'EXTENSION DE CODE

- CE06      CSI      Introducteur de séquence de commande.  
 Fonction: donner des représentations ad-  
 ditionnelles, en particulier des fonctions  
 des commandes avec paramètres (fonction de  
 présentation).



#### 4 - COMMANDE DIVERSE

- CM04      IGS      Identification de sous-répertoire de caractères graphiques:
- Fonction avec paramètre graphique, utilisée pour signaler au terminal de réception qu'un sous-répertoire particulier du répertoire complet de caractères graphiques doit être utilisé dans le texte qui suit.
- La valeur du paramètre identifie le sous-répertoire conformément au registre de sous-répertoire. Par défaut, on a le répertoire de base. Si un sous répertoire a été identifié, il doit être répété avant le premier texte sur chaque page.

#### C - LES SOUS-RÉPERTOIRES

doivent contenir les caractères graphiques suivants:

- a - les 26 lettres minuscules a ...z.
- les 26 lettres majuscules A ...Z.
- b - les 10 chiffres décimaux 0 ...9.
- c - des caractères graphiques non alphabétiques  
    ' ( ) , - . / : ? + =
- d - plus éventuellement d'autres caractères.

#### D - FONCTIONS DE COMMANDE

Représentation CSI     $P_1 \dots P_m \quad I_1 \dots I_m . F.$

- CSI est représenté par 9/11 dans le code 8 éléments  
    (Dans le code à 7 moments, CSI est représenté par ESC 5/11.)
- $P_1 \dots P_m$ : combinaison d'éléments binaires (chiffres)



représentant les valeurs des paramètres (à omettre si la fonction de commande n'a pas de paramètre ou pouvant être omis si la valeur par défaut peut être applicable).

- $I_1 \dots I_m$  combinaison d'éléments binaires de la colonne 2 qui avec la combinaison finale F identifient la fonction de commande.
- F. est une combinaison d'éléments binaire 4/0 à 7/14 (à l'exception de la position 7/15) qui conclut la séquence de commande.

Toute combinaison binaire qui ne correspond pas aux formes définies ci-dessus correspond à un état d'erreur.

#### Représentation des paramètres

La chaîne de combinaisons d'éléments binaires  $P_1 \dots P_m$  contenue dans une séquence de commande est appelée chaîne de paramètre.

#### Format de la chaîne

- constituée d'une ou plusieurs sous-chaîne de paramètre dont chacune représente une valeur du paramètre.
- chaque sous-chaîne est constituée d'une combinaison d'éléments représentant les chiffres décimaux.
- les sous-chaînes sont séparées par; (3/11).

### E - EXEMPLE: FONCTION DE PRÉSENTATION

- On désire: une page de largeur de base avec
  - espacement horizontal entre caractère: 10 caractères / 25,4 mm
  - espacement vertical entre caractère: 3 lignes / 25,5 mm

la commande sera:

CSI 1; 0; 2    2/0 4/10    2/0 4/11    2/0 4/12  
                          PLS                           SHS                           SVS

[ 9/11 1; 0; 2    2/0 4/10    2/0 4/11    2/0 4/12  
 code 8 moments

[ ESC 5/11 1; 0; 2    2/0 4/10    2/0 4/11    2/0 4/12  
 code à 7 moments

≡ ESC [ 1; 0; 2 ! J    ! K    ! L

- Par défaut de paramètres, on aurait:

- une page de hauteur de base
- espacement vertical en caractère: 6 lignes/25,4 mm
- espacement horizontal en caractère: 10 caractères/  
25,4 mm

CSI    2/0 4/10    2/0 4/11    2/0 4/12



## ANNEXE D5

L A O 1

pour les caractères alphabétiques:

chiffre impair = lettre minuscule

chiffre pair = lettre majuscule

aucune signification particulière quand la première lettre est C, N ou S

pour les caractères alphabétiques:

0 = lettre sans signe diacritique

1 à 3 = lettre avec signe diacritique au-dessus

4 = lettre avec signe diacritique au-dessous

5 = ligature

6 = forme spéciale

aucune signification particulière quand la première lettre est C, N ou S

pour les caractères alphabétiques:

A à Z = lettre correspondante de l'alphabet latin ou translittération dans cet alphabet d'une lettre d'un autre alphabet

Si la première lettre est C:

E = fonction de commande pour extension de code

F = fonction de commande pour mise en page

P = fonction de commande pour présentation

M = autre fonction de commande

Si la première lettre est N:

D = chiffre décimal

F = fraction

S = texte en inférieur ou en supérieur

Si la première lettre est S:

A = signe arithmétique

C = symbole monétaire

D = signe diacritique

P = signe de ponctuation

M = autre symbole

L = caractère de l'alphabet latin

C = fonction de commande

N = caractère graphique non alphabétique

S = caractère graphique spécial

Système d'identification

## A N N E X E    E

### PRIMITIVES ET DIAGRAMMES D'ÉTATS DU NIVEAU PRÉSENTATION

=====

Nous reprenons dans cette annexe les primitives et les paramètres de niveau présentation. Ceux-ci ont déjà été évoqués au chapitre V, paragraphe 3.3.3.

On trouvera dans les feuilles qui suivent:

- E.1 - La liste des primitives et leurs paramètres.
- E.2 - La liste des commandes avec en regard les réponses qu'elles suscitent.
- E.3 - Le graphe correspondant au diagramme d'états et de transitions: côté appelant.
- E.4 - Le graphe correspondant au diagramme d'états et de transitions: côté appelé.
- E.5 - Les tables séquentielles correspondant aux annexes E.3 et E.4.

Le lecteur intéressé par une description détaillée de ces primitives et de leurs paramètres pourra trouver des renseignements supplémentaires dans le "Projet Avis S-62" du CCITT.



PGI et PI pour les éléments de procédure du document

<u>DOCUMENT</u>  Identificateur de commande	Identificateurs de groupe de paramètres (PGI)		Identificateurs de paramètres (PI)	
	Description	Obligatoire ou non obligatoire	Description	Obligatoire ou non obligatoire
CDC			Numéro de référence du document	obligatoire
			Identificateur d'interfonctionnement de services	non obligatoire
			Identificateur de type de document	non obligatoire
	Capacités non essentielles du terminal	non obligatoire	Jeux de caractères graphiques	non obligatoire
			Jeux de caractères de commande	non obligatoire
			Formats de page	non obligatoire
			Capacités diverses du terminal	non obligatoire
	Usage privé	non obligatoire		
CDC	Raccordement des documents	obligatoire	Numéro de référence du document	obligatoire
			Numéro de référence du point de repère	obligatoire
	(Obligatoire seulement pour le raccordement dans un nouvel échange)		Identificateur du terminal demandé	obligatoire
			Identificateur du terminal demandeur	non obligatoire
			Numéro de référence d'échange supplémentaire	non obligatoire

			Date et heure	obligatoire
			Numéro de référence d'échange supplémentaire	non obligatoire
			Identificateur d'interfonctionnement de services	non obligatoire
			Identificateur de type de document	non obligatoire
			Référence de document (échange en cours)	obligatoire
			Autres paramètres de CDS	non obligatoire
	Usage privé	non obligatoire	_____	_____
CDE			Numéro de référence du point de repère	obligatoire
CDR			Raison (document)	non obligatoire
CDJ			Raison (document)	non obligatoire
CDPB			Numéro de référence du point de repère	obligatoire
CDCL	Capacités non essentielles du terminal	non obligatoire	Négociation de la capacité mémoire	non obligatoire
			Jeux de caractères graphiques	non obligatoire
			Jeux de caractères de commande	non obligatoire
			Formats de page	non obligatoire



			Capacités diverses du terminal	non obligatoire
	Usage privé	non obligatoire	_____	_____
CDJI			_____	_____
CDRP			Numéro de référence du point de reprise	obligatoire
CDRPR			Numéro de référence du point de reprise	obligatoire
RDEP			Numéro de référence du point de repère	obligatoire
RDRP			_____	_____
RDCP			_____	_____
RDPBP			Numéro de référence du point de repère	obligatoire
			Capacité de réception compromise	obligatoire
RDPBN			Raison (document)	obligatoire
RDCR			Acceptation des paramètres CDCL	non obligatoire
	Capacités non essentielles du terminal	non obligatoire	Négociation de la capacité mémoire	non obligatoire
			Jeux de caractères graphiques	non obligatoire
			Jeux de caractères de commande	non obligatoire

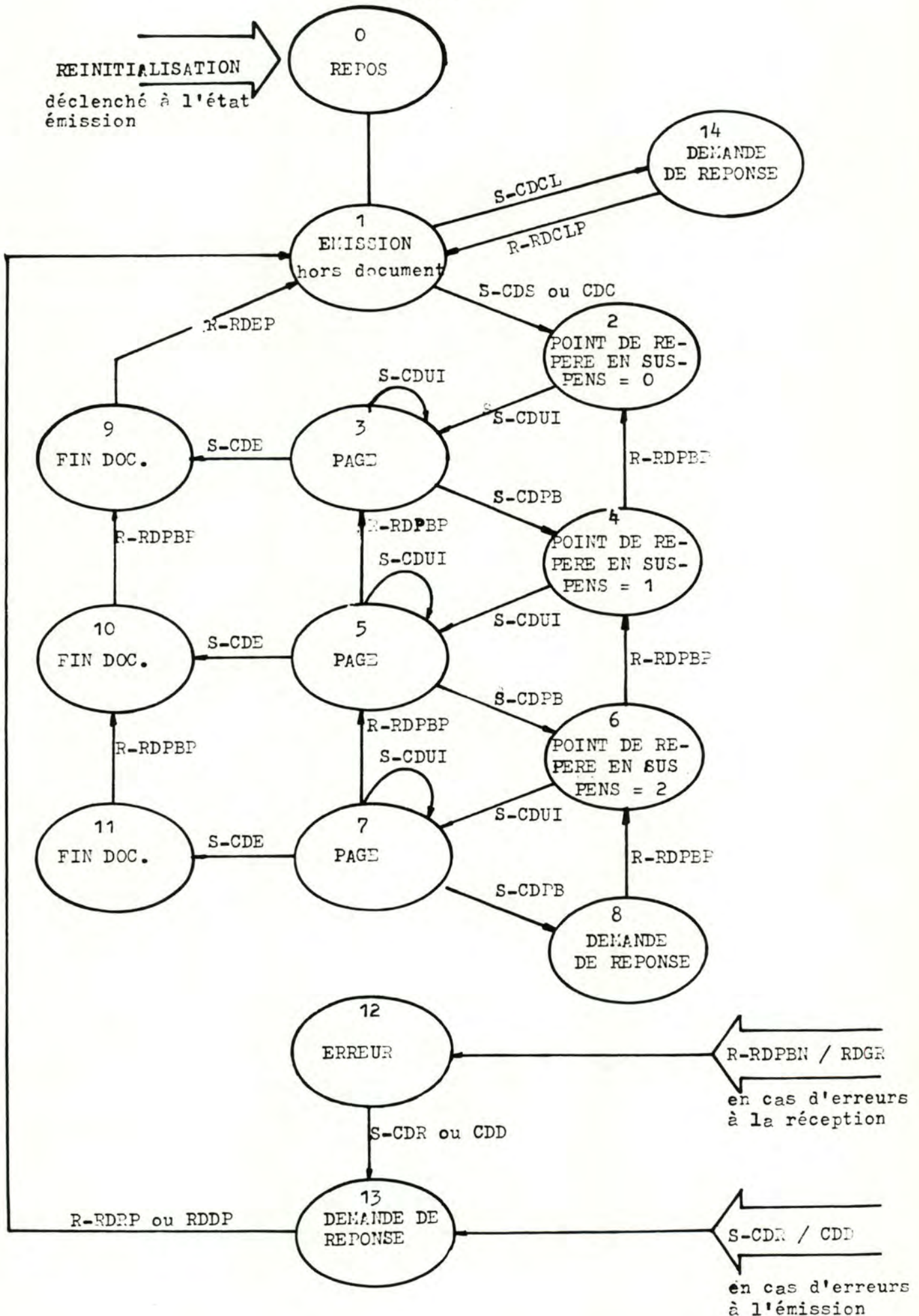
<u>DOCUMENT</u>	PGI		PI	
Identificateur de réponse	Description	Obligatoire ou non obligatoire	Description	Obligatoire ou non obligatoire
			Formats de page	non obligatoire
			Capacités diverses du terminal	non obligatoire
	Usage privé		_____	_____
RDGR			Indication de valeurs de paramètre	obligatoire
RDRPN			Numéro de référence du point de reprise	obligatoire
RDEPR			Numéro de référence du point de reprise	obligatoire



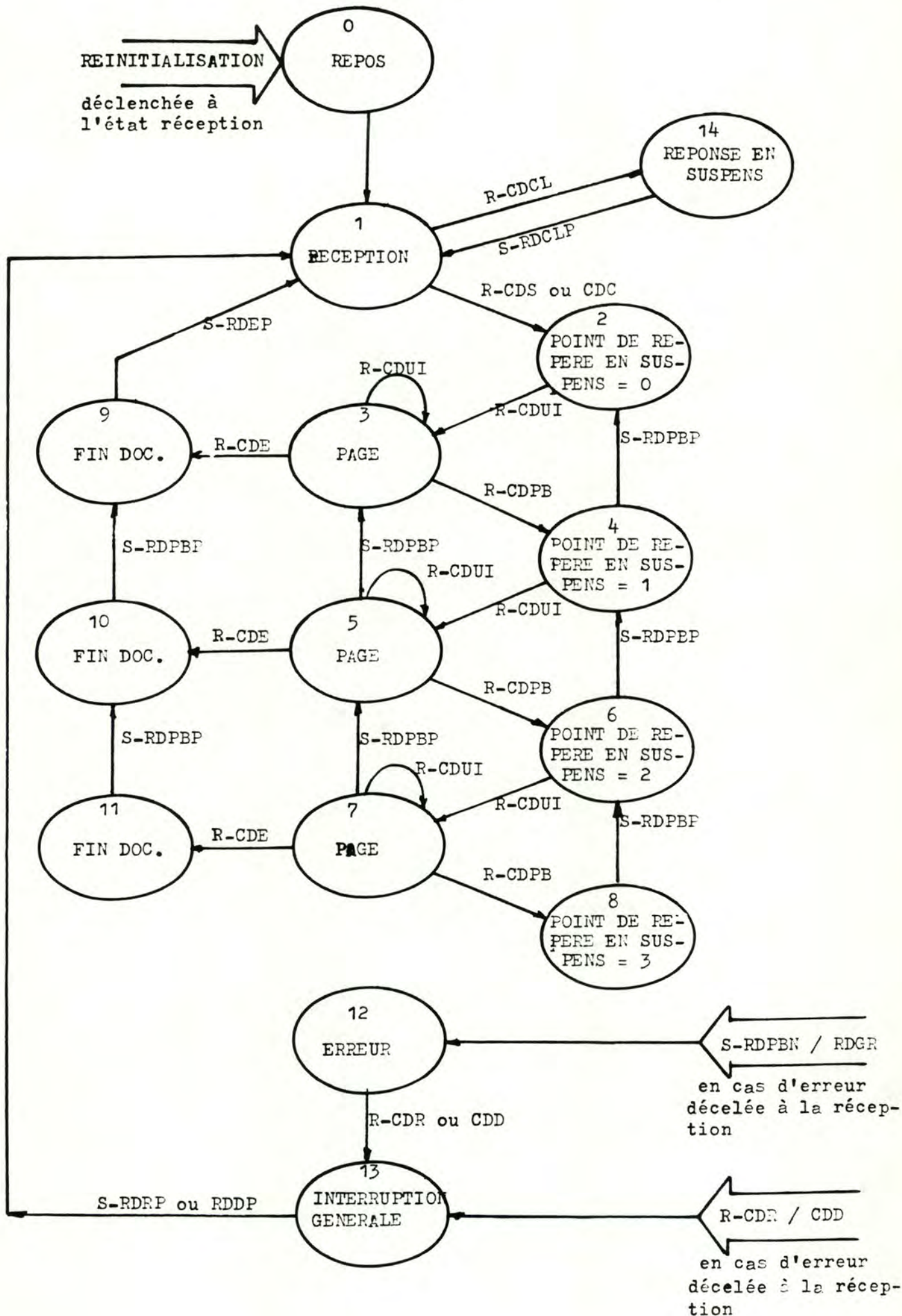
## ANNEXE E2

Appelant		Appelé
Contrôle de document		
début	CDS →	
continuation	CDC →	
liste des capacité	CDCL →	
		← RDCLP (OK)
fin	CDE →	
		← RDEP (OK)
rejet	CDD →	
		← RDDP (OK)
synchronisation	CDR →	
		← RDRP (OK)
Transfert d'informations		
information	CDUI →	
Recouvrement d'erreurs		
		← RDGR (rejet total du document)
limite de page	CDPB →	
		← RDPBP (OK)
		← RDPBN (OK)

LISTE DES COMMANDES ET REPONSES AU NIVEAU PRESENTATION.











A P P E L E

Description	Phase	I N I T I A L	C D S / C D C	Reçu			C D R / C D D	C D C L	A V O R T E	
Repos	0	1								
Réception	1		2				14	18	12	RDGR
Point de repère 0.	2			3				18		
Page	3			3	4	9		18		
Point de repère 1.	4			5				18	12 2	RDPBN RDPBP
Page	5			5	6	10		18	12 3	RBPBN RDPBP
Page										
Point de repère 2.	6			7				18	12 4	
Page	7			7	8	9		18	12 5	
Point de repère 3.	8							18	12 6	
Fin-doc	9								1	RDEP
Fin-doc	10								9 12	RDPBP RDPBN
Fin-doc	11								10 12	RDPBP RDPBN
Erreur	12							18		
Interruption gen.	13								1 1	RDDP RDRD
Réponse	14								1	RDCLP
Avorté	18						13			

## A N N E X E    F

### PROTOCOLE D'APPAREIL VIRTUEL

On trouvera dans cette annexe différents renseignements concernant le protocole d'appareil virtuel dont il a été question au chapitre V - paragraphe 3.3.5.

Les feuilles suivantes comporteront:

- F.1 - Les primitives du protocole PAV
  - Primitives de texte
  - Primitives de négociation
  - Primitives télégrammes d'interruption
  - Primitives marqueurs d'interruption
- F.2 - Les caractéristiques indépendantes de la notion de classe de terminal virtuel.
- F.3 - Les caractéristiques liées à la notion de classe de terminal virtuel.
- F.4 - Protocole d'appareil virtuel.



## A N N E X E    F 1

### LES PRIMITIVES DU PROTOCOLE PAV

#### 1 - PRIMITIVES DE TEXTE:

Dans cette classe, sont regroupées:

- les chaînes de caractères,
- les fonctions d'adressage (par exemple NL, FORWARD TAB, NEXT FIELD, CURSOR ADDRESS),
- les fonctions de description de l'écran (par exemple attributs et fonctions d'effacement).

#### 2 - PRIMITIVES DE NEGOCIATION:

Elles restent à définir dans la mesure où le choix du schéma de négociation (symétrique ou dissymétrique) n'a pas encore été fixé.

Elles ont fait l'objet d'études d'une part des sociétés dites savantes comme ISO, IFIP,... incitant les Universités de concrétiser les principes théoriques émis par leurs commissions; d'autre part, des organismes internationaux comme C.C.I.T.T., ECMA, AFNOR... dont la priorité était donnée aux propositions rapides de primitives fiables.

La synthèse minimal rassemble finalement les primitives proposées par l'organisme le plus important, le C.C.I.T.T.

#### 3 - PRIMITIVES DE CONTROLE D'UNITES AUXILIAIRES:

Non discutées ici.

#### 4 - PRIMITIVES-TELEGRAMMES D'INTERRUPTION:

Ce sont les télégrammes utilisés lors du traitement d'une condition d'interruption. On peut distinguer les télégrammes de commande (émis par le processus générateur de la condition d'interruption: T-INTERRUPT) et les télégrammes de réponse (émis par le processus récepteur: T-PURGE ou T-RESUME).

## 5 - PRIMITIVES-MARQUEURS D'INTERRUPTION:

Nous avons souligné l'existence de primitives séquencées (i.e. non prioritaires) appelées marqueurs (cfr Annexe 4 consacrée au traitement d'une condition d'interruption).

Nous mentionnons également l'existence de deux types d'entités au niveau  $VL_2$ :

- les blocs,
- les télégrammes.

La liaison entre ces deux types d'entités et les cinq catégories de primitives que nous venons de faire apparaître est la suivante:

- a - pour les blocs: un bloc est constitué
  - soit d'une ou plusieurs primitives de texte,
  - soit d'une primitive de négociation,
  - soit d'une primitive de contrôle d'unités auxiliaires,
  - soit d'une primitive-marqueur d'interruption.
- b - pour les télégrammes: un télégramme se compose d'une primitive-télégramme d'interruption.



## A N N E X E    F 2

### CARACTÉRISTIQUES INDÉPENDANTES

#### DE LA NOTION DE CLASSE DE TERMINAL VIRTUEL

##### A - MODE DE GESTION:

- mode réel
- mode virtuel.

En mode réel, le terminal réel est pris en charge comme tel c'est-à-dire avec toutes les caractéristiques propres à son langage) par l'application via le réseau.

Dans ce mode, le rôle joué par le module d'adaptation est assez élémentaire:

- aider à l'établissement et à la fermeture de la liaison virtuelle  $VL_2$  (voir 53.3.5.1)
- assurer la bufferisation des données en provenance du terminal réel et leur expédition vers l'application;
- analyser le flux de caractères issu du terminal réel pour y détecter le caractère spécial utilisé localement pour passer en mode commande (ce caractère n'est pas transmis vers l'application). Le mode commande est un mode permettant à l'opérateur de dialoguer avec le module d'adaptation fonctionnant en interpréteur de commandes, ce dialogue restant invisible pour l'application.

Il permet entre autres à l'opérateur d'interrompre l'échange de données avec l'application pour formuler une demande de fermeture de la liaison virtuelle à laquelle le terminal participe.

En mode virtuel, le module d'adaptation fait en sorte que les primitives transmises sur VL<sub>2</sub> aient une représentation standard au niveau du réseau.

B - JEU ET CODE DE CARACTERES:

- IA5
- APL

La possibilité d'extension à d'autres codes (EBCDIC, jeux de caractères spéciaux) n'est pas abordée.

C - MODE D'ECHANGE DES DONNEES:

- échange alterné
- échange libre.

D - DIMENSIONS DE LA PAGE:

- nombre de positions par ligne
- nombre de lignes par page

E - EXISTENCE D'UNITES AUXILIAIRES:

L'étude détaillée des unités auxiliaires n'est pas abordée ici.

F - NOMBRE DE FONCTIONS PROGRAMME ET DE TOUCHES  
D'INTERRUPTION



## A N N E X E    F 3

---

### CARACTÉRISTIQUES LIÉES

---

---

#### À LA NOTION DE CLASSE DE TERMINAL VIRTUEL

---

---

##### A - CLASSE DU TERMINAL VIRTUEL :

- SCROLL-MODE
- PAGE MODE
- DATE ENTRY

##### B - FONCTIONS D'ADRESSAGE :

NEW LINE, FORWARD TAB, HOME, CURSOR ADDRESS, NEXT FIELD...

##### C - ATTRIBUTS :

- contrôle d'accès
  - Protégé/non protégé
  - numérique/alphanumérique
  - sélection par crayon lumineux
  - action du crayon lumineux
- contrôle de visualisation
  - affichable/non affichable
  - affichable normal/affichable contrasté

##### D - FONCTIONS D'EFFACEMENT :

- CLEAR

## A N N E X E    F 4

### PROTOCOLE D'APPAREIL VIRTUEL

Le protocole du terminal virtuel dispose de mécanismes qui concernent:

- le contrôle du mode de dialogue (échange libre ou alterné),
- la phase de négociation des caractéristiques du terminal virtuel,

auxquels il faut adjoindre:

- le traitement des conditions d'interruption.

#### a - Contrôle du mode de dialogue

Le dialogue entre les deux stations VTP sur la liaison virtuelle  $VL_2$  peut s'effectuer selon deux schémas d'échange: l'échange libre ou l'échange alterné. Dans le cas de l'échange alterné, une station VTP cède la parole à son correspondant de manière explicite à l'aide d'un "YOUR-TURN". Elle ne pourra reprendre la transmission de données que lorsqu'elle aura elle-même reçu un "YOUR-TURN" de son correspondant. Le mode de dialogue utilisé et, dans le cas de l'échange alterné, l'attribution du premier tour de parole seront fixés au cours de la phase de négociation.

#### b - Phase de négociation des caractéristiques du terminal virtuel

Dans ce paragraphe, nous allons décrire deux schémas de négociation possibles. Nous donnerons ensuite la liste détaillée des caractéristiques négociables.



### SCHEMA 1 (SCHEMA DISSYMETRIQUE)

Ce schéma a été introduit dans le protocole terminal virtuel de TELENET.

Deux cas peuvent se présenter:

- a - Le premier est celui où l'application présente une certaine flexibilité qui lui permet de s'adapter à certaines caractéristiques du terminal virtuel. L'échange des primitives "DEMANDE DE POSSIBILITES" et "VOICI MES POSSIBILITES" (voir fig. 1) permet précisément à l'application de prendre connaissance des caractéristiques du terminal virtuel. La négociation s'achève par l'échange des deux primitives "VOICI NOS PARAMETRES" et "REPONSE" permettant à l'application d'exprimer ses desiderata et au terminal virtuel d'exprimer son accord ou son désaccord.

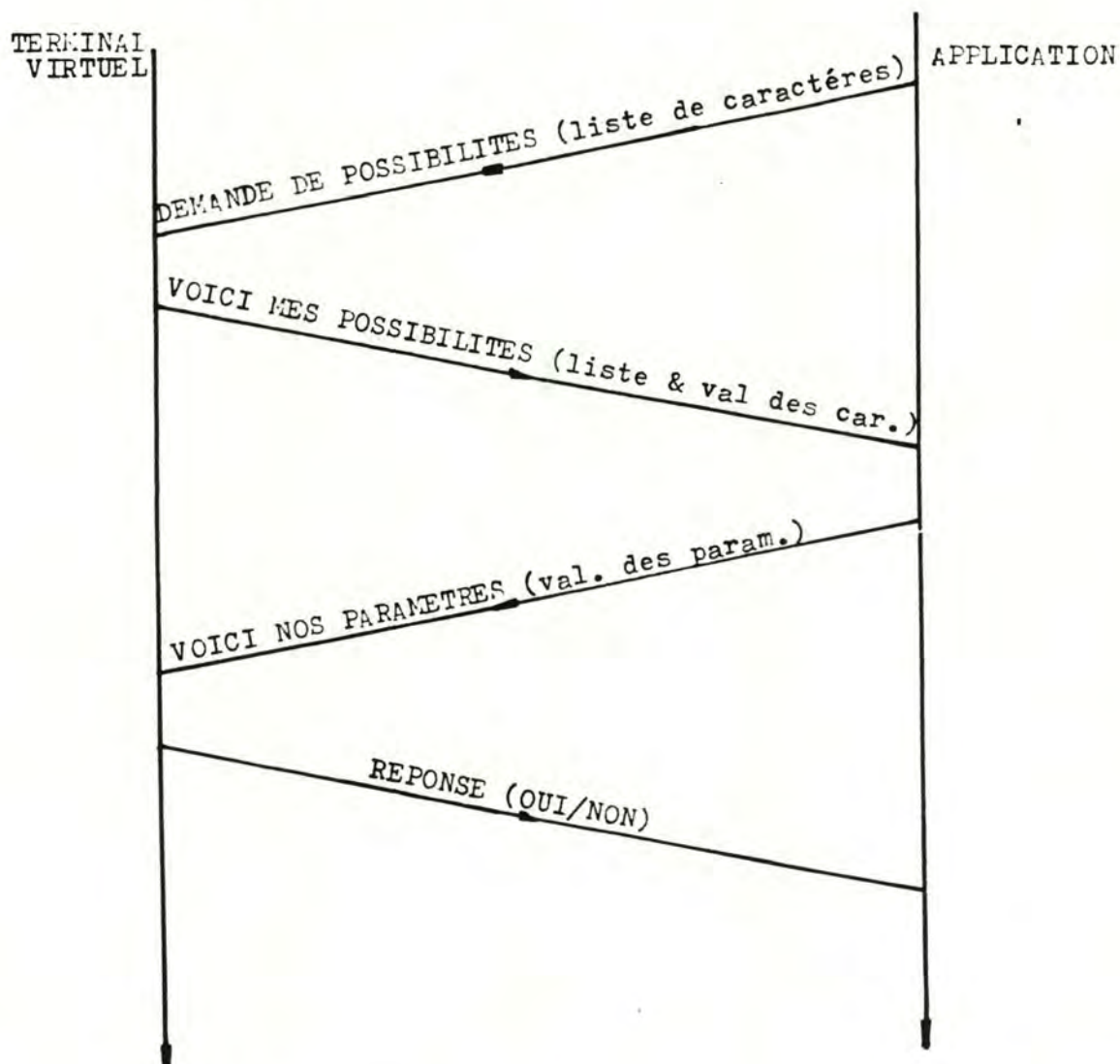


Figure 1 - Schéma de négociation dissymétrique (1er cas)

- b - Le deuxième cas est celui où l'application ne désire pas (ou n'a pas la possibilité de) s'adapter aux caractéristiques du terminal virtuel.
- Dans ce cas, la négociation se réduit à l'échange des deux primitives "VOICI NOS PARAMETRES" et "REPONSE" (voir fig. 2).

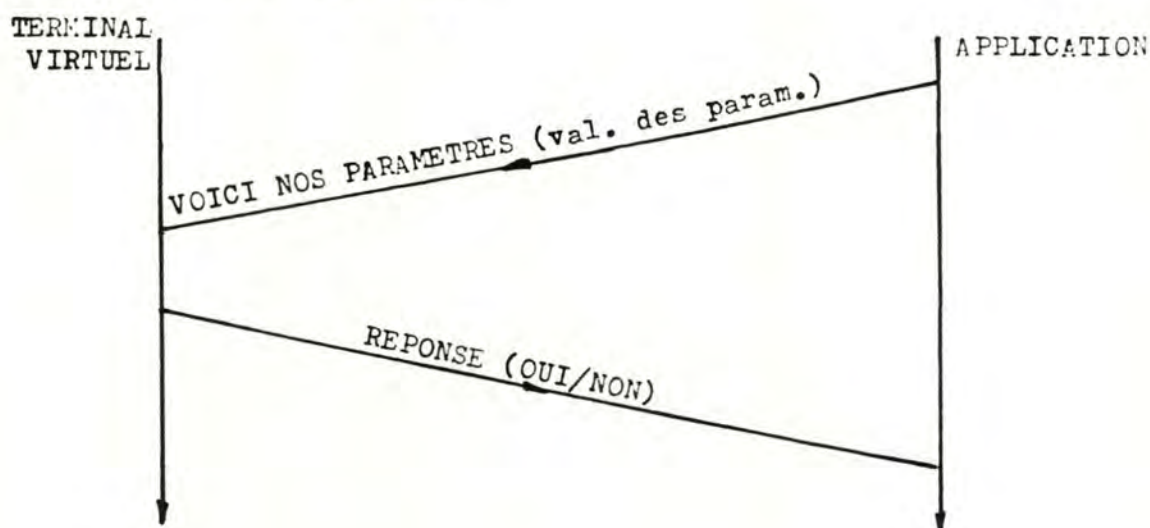


Fig. 2 - Schéma de négociation dissymétrique (2ème cas).

Ce schéma de négociation convient aussi bien pour la négociation initiale que pour une renégociation en cours de dialogue.

#### SCHEMA 2 (SCHEMA SYMETRIQUE)

Ce schéma a été proposé par l'équipe EIN de Zurich (P. SCHICKER et A. DUENKI).

Au contraire du schéma 1, le schéma de négociation proposé par SCHICKER et DUENKI est parfaitement symétrique. Son principe est le suivant: chaque partie envoie à son correspondant une primitive D-POSSIBILITES dans laquelle elle définit ses propres possibilités (sous forme d'une liste de caractéristiques avec leurs valeurs) (voir fig. 3).



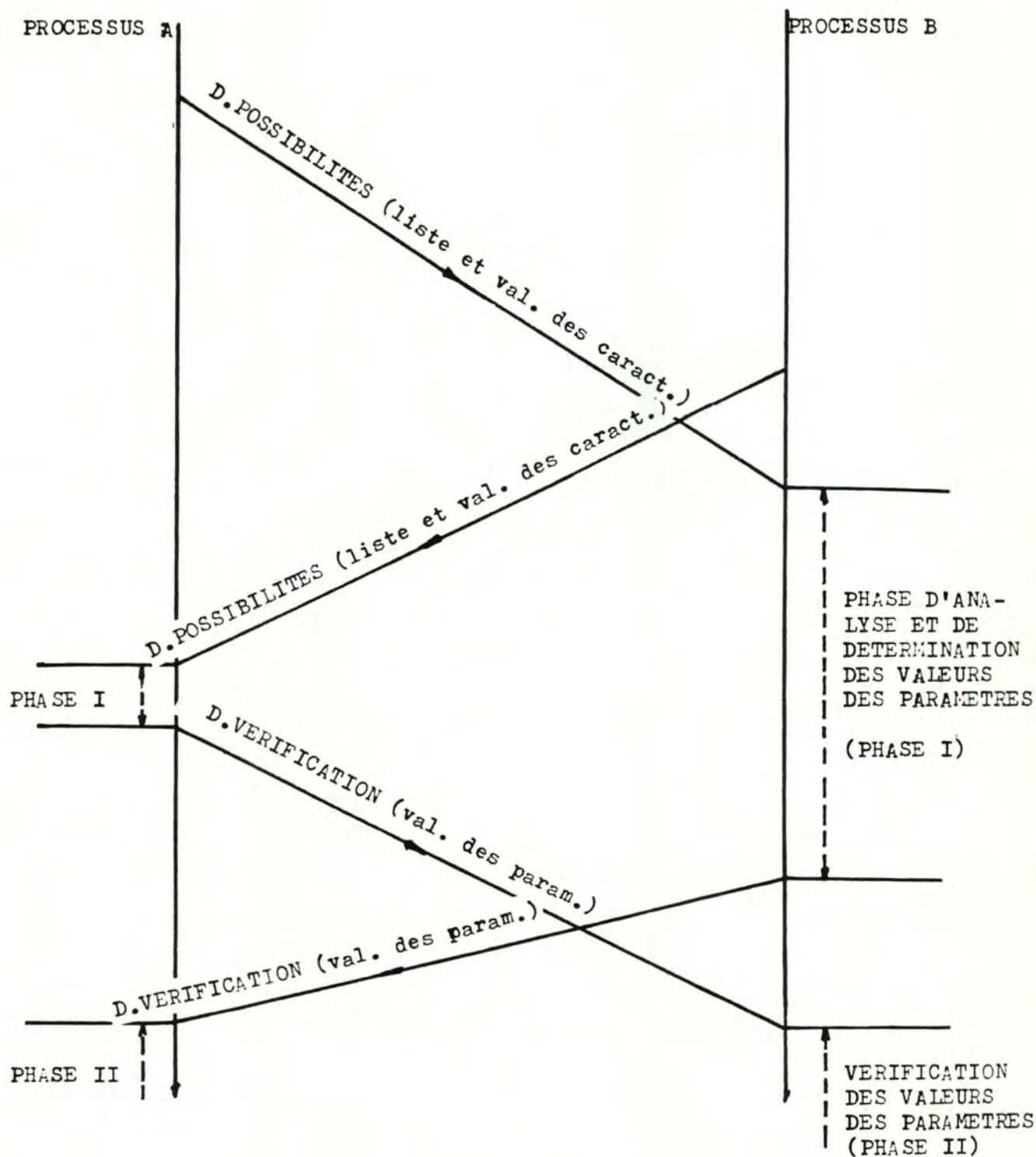


Figure 3 - Schéma de négociation symétrique.

Sur réception de la primitive D-POSSIBILITES émise par son correspondant, chacune des deux parties examine individuellement si les deux listes de possibilités (la sienne et celle que son correspondant lui a transmise) sont compatibles de façon à construire une liste de paramètres communs. Il faut évidemment obtenir que les listes de paramètres auxquelles les deux parties aboutissent grâce à l'examen qu'elles effectuent indépendamment, concordent. Dans ce but, et afin de lever certaines indéterminations pouvant résulter de la comparaison des deux primitives D-POSSIBILITES, il est nécessaire de définir des règles de priorité (dans la négociation du mode d'échange, on peut par exemple admettre la priorité de l'échange libre sur l'échange alterné).

De toute façon, ces règles de priorité doivent être définies une fois pour toutes et acceptées par l'ensemble des applications et terminaux virtuels appelés à communiquer via le réseau.

A titre de vérification, SCHICKER et DUENKI proposent de faire suivre l'échange des primitives D-POSSIBILITES par un échange des primitives D-VERIFICATION grâce auxquelles chacune des deux parties peut communiquer à l'autre la liste de paramètres à laquelle elle a abouti.



Les schémas de négociation symétrique et dissymétrique présentent chacun leurs avantages et leurs inconvénients. Nous en avons fait la liste dans le tableau ci-dessous.

	Schéma dissymétrique (sch.1)	Schéma symétrique (sch.2)
<u>Avantages de la méthode</u>	<ul style="list-style-type: none"> <li>- la transmission sur VL2 de la liste complète des caractéristiques n'est pas toujours nécessaire.</li> <li>- les algorithmes de décision (détermination des paramètres) sont définis uniquement de manière locale.</li> </ul>	<ul style="list-style-type: none"> <li>- le schéma est applicable aussi bien dans le cas d'une communication entre processus identiques (terminal-terminal ou application-application que dans celui de processus différents (terminal-application).</li> </ul>
<u>Inconvénients de la méthode</u>	<ul style="list-style-type: none"> <li>- dans le cas d'une communication entre processus identiques (terminal-terminal ou application-application), il est nécessaire de réintroduire artificiellement une dissymétrie (p. ex. notion d'appelant-appelé)</li> </ul>	<ul style="list-style-type: none"> <li>- il est nécessaire d'envoyer sur VL<sub>2</sub> la liste complète des caractéristiques.</li> <li>- le schéma suppose l'existence de critères de décision définis au niveau du réseau</li> </ul>

Avantages et inconvénients des 2 schémas de négociation

### c - Traitement des conditions d'interruption

Le traitement des conditions d'interruption que nous proposons dans ce travail est très général en ce sens que nous avons cherché à définir un schéma de traitement des conditions d'interruption unique mais susceptible de convenir quel que soit le comportement du récepteur de la condition d'interruption. Ce comportement doit être considéré comme inconnu au niveau  $VL_2$ , car il fait partie d'un protocole de niveau supérieur (conventions entre opérateur et application).

Aussi le schéma de traitement doit-il offrir au récepteur de la condition d'interruption:

- la possibilité d'opter soit pour l'élimination des données qu'il a émises et qui sont encore en transit dans le réseau, soit pour la poursuite d'une transmission normale de celle-ci. Du côté de l'émetteur de la condition d'interruption, ceci implique une inhibition momentanée de la réception des données dans l'attente d'une information sur le choix opéré par le récepteur.
- la possibilité de s'attribuer le premier tour de parole à l'issue de la phase de traitement d'interruption ou de le céder à l'émetteur de la condition d'interruption. Ceci correspond donc à une renégociation du tour de parole et offre un intérêt évident lorsque le dialogue est organisé selon le mode d'échange alterné. En effet, une application peut désirer reprendre la parole (cas d'une écriture après interruption) ou se mettre à l'écoute du correspondant (cas d'une lecture après interruption). Il se peut que le tour de parole qui prévalait au début de la phase d'interruption ne convienne plus lors de la reprise de l'échange de données.



Le schéma de traitement que nous proposons permettra par conséquent de couvrir des situations variées telles INTERRUPT PROCESS, ABORD OUTPUT ou PLEASE (permettant à un processus de reprendre la parole au cours d'un échange alterné).

Lorsqu'un processus veut transmettre une condition d'interruption à son correspondant, deux primitives sont envoyées sur  $VL_2$ .

1 - Une primitive d'interruption (télégramme)

Cette primitive permet d'avertir le correspondant le plus rapidement possible de l'existence d'une condition d'interruption;

2 - Une primitive séquencée (bloc) que nous nommerons marqueur:

Cette primitive est utilisée pour indiquer au correspondant que l'échange de données peut reprendre normalement selon un tour de parole précisé dans un champ du marqueur.

Sur réception d'une condition d'interruption, le processus récepteur procède à une analyse de la condition d'interruption et répond par un double envoi sur  $VL_2$ :

1 - un télégramme qui précise s'il y a lieu ou non de procéder à l'élimination des données encore en transit et qui redistribue le tour de parole;

2 - un marqueur servant à indiquer au correspondant à quel moment le télégramme d'interruption a été reçu.

## A N N E X E    G

### PROTOCOLE STANDARD X.25

=====

On trouve dans cette annexe des renseignements complémentaires à ceux donnés au chapitre VI paragraphe 2.

Les feuilles suivantes contiennent les descriptions

- G.1 - des paquets nécessaires à l'établissement d'une communication.
- G.2 - des paquets nécessaires à la libération d'une connexion.
- G.3 - des paquets nécessaires au transfert de données et au contrôle de flux.
- G.4 - des paquets de reprises, de réinitialisation, d'interruption avec leur confirmation.
- G.5 - des formats des différents paquets sous la forme d'un tableau.

On peut trouver des renseignements complémentaires dans les publications du CCITT.



## ANNEXE G 1

### PAQUET D'APPEL ET APPEL ENTRANT

		8	7	6	5	4	3	2	1
OCTET	1	0 0 0 1				NUMERO			
	2	DE VOIE LOGIQUE							
	3	0 0 0 0				1 0 1 1			
	4	Long. adresse ETTD appelant				Long. adresse ETTD appelé			
		ADRESSE D'ETTD							
						0 0 0 0			
		0 0		Longueur du champ de facilités					
		FACILITES							
		DONNEES D'APPEL DE L'UTILISATEUR							

Les longueurs d'adresse sont exprimées en compte de demi-octets      bit 5 et bit 1 = bit de poids faible.

Le champ adresse de l'ETTD comprend dans l'ordre:

- l'adresse de l'ETTD appelé
- l'adresse de l'ETTD appelant

Chaque chiffre décimal d'une adresse est codé sur un quartet en binaire (bit 5 et 1 poids faible) en commençant par le chiffre de poids fort (codé dans le quartet (8, 7, 6, 5) de l'octet 5).

Longueur du champ de facilité: Ce champ n'est présent que lorsqu'un ETTD désire utiliser les services complémentaires:

- taxation au demandé
- sélection des paramètres de contrôle de flux
- groupe fermé d'abonnés

Il est composé d'une suite de paires d'octets. Le 1er octet caractérise le type de service et le 2ème précise le service.

1 - <u>Taxation au demandé</u>	bit	8	7	6	5	4	3	2	1
	1er octet	0	0	0	0	0	0	0	1
	2ème -	0	0	0	0	0	0	0	a

a = 1 si la taxation au demandé est prescrite

2 - <u>Sélection des paramètres de contrôle de flux</u>	bit	8	7	6	5	4	3	2	1
	1er octet	0	0	0	0	0	0	1	0
	2ème -	<u>m / n</u>							

m précise la classe de débit désirée pour la transmission depuis l'appelé et n depuis l'appelant suivant le code:

classe de débit en bits/s	bit	8 7 6 5			
		ou 4 3 2 1			
0		0	0	0	0
5 0		0	0	1	1
1 1 0					
1 5 0		0	1	0	0
3 0 0		0	1	0	1
6 0 0		0	1	1	0
1 2 0 0		0	1	1	1
2 4 0 0		1	0	0	0
4 8 0 0		1	0	0	1
9 6 0 0		1	0	1	0
1 9 2 0 0		1	0	1	1



3 - Groupe fermé d'abonnés

bit	8	7	6	5	4	3	2	1
ler octet	0	0	0	0	0	0	1	1

Le 2ème octet précise le numéro du groupe considéré pour la communication.

Données d'appel de l'utilisateur: ce champs est à la disposition de l'ETTD appelant et lui permet de transmettre des informations d'appel (par exemple un mot de passe) quelconques à l'appelé. Sa taille maximale est de 16 octets.

## PAQUET COMMUNICATION ACCEPTEE

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1	NUMERO			
2	DE VOIE LOGIQUE							
3	0	0	0	0	1	1	1	1

## ANNEXE G 2

### PAQUET DEMANDE DE LIBERATION ET INDICATION DE LIBERATION

		8	7	6	5	4	3	2	1
OCTET	1	0 0 0 1				NUMERO			
	2	DE VOIE LOGIQUE							
	3	0	0	0	1	0	0	1	1
	4	CAUSE DE LIBERATION							

Le codage du champ cause de libération est le suivant:

	8	7	6	5	4	3	2	1
Libération par l'ETTD	0	0	0	0	0	0	0	0
Numéro occupé	0	0	0	0	0	0	0	1
Dérangement	0	0	0	0	1	0	0	1
Erreur de procédure distante	0	0	0	1	0	0	0	1
Numéro refusant la taxation au demandé	0	0	0	1	1	0	0	1
Appel non valide	0	0	0	0	0	0	1	1
Interdiction d'accès	0	0	0	0	1	0	1	1
Erreur de procédure locale	0	0	0	1	0	0	1	1
Saturation du réseau	0	0	0	0	0	1	0	1
Numéro inconnu	0	0	0	0	1	1	0	1

Nota: .Appel non valide = demande de service complémentaire non valable

.Interdiction d'accès = l'ETTD appelant n'est pas autorisé à établir une communication avec l'ETTD appelé.  
Par exemple pour des motifs de groupe fermé d'abonnés.

.Dérangement: causes de dérangement de l'abonné appelé:

- ETTD hors service
- défaut sur sa liaison de raccordement

### PAQUET CONFIRMATION DE LIBERATION

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1	NUMERO			
2	DE VOIE LOGIQUE							
3	0	0	0	1	0	1	1	1



## A N N E X E    G 3

### PAQUET DE DONNEES

		8	7	6	5	4	3	2	1
OCTET	1	Q	0	0	1	NUMERO			
	2	DE VOIE LOGIQUE							
	3	P (R)		M	P (S)		0		
		DONNEES DE L'UTILISATEUR							

Le bit M (marque de données à suivre) indique la "fragmentation" d'un message en paquets (M - 0 signifie: dernier paquet du message).

Le bit de qualification, permet à l'ETTD de distinguer des données de l'usager et des informations de supervision transmises sur le même circuit virtuel.

Le numéro P(S): les paquets de données émis sont numérotés cycliquement (module 8) au moyen de P(S)

Le numéro P(R): le récepteur indique au moyen de P(R) les numéros des paquets qu'il accepte de recevoir de l'émetteur.

### PAQUETS DE CONTROLE DE FLUX

Paquet RR (Ready Receive)

		8	7	6	5	4	3	2	1
OCTET 1		0	0	0	1	NUMERO			
2	DE VOIE LOGIQUE								
3		P (R)			0	0	0	0	1

Paquet RNR (Ready not receive)

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1	NUMERO			
	DE VOIE LOGIQUE							
	P (R)			0	0	1	0	1

## ANNEXE G 4

### PAQUET DE REPRISE ET INDICATION DE REPRISE

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1	NUMERO			
2	DE VOIE LOGIQUE							
3	0	0	0	1	1	0	1	1
4	CAUSE DE REINITIALISATION							
5	CODE DE DIAGNOSTIC							

#### Champ cause de reprise:

- dans les paquets DEMANDE DE REPRISE, il est à zéro
- dans les paquets INDICATION DE REPRISE, il a la signification suivante:

bit	8	7	6	5	4	3	2	1
erreur de procédure locale	0	0	0	0	0	0	0	1
saturation du réseau	0	0	0	0	0	0	1	1

### PAQUET CONFIRMATION DE REPRISE

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1



# PAQUET DE DEMANDE DE REINITIALISATION ET INDICATION DE REINITIALISATION

		8	7	6	5	4	3	2	1
OCTET	1	0	0	0	1	0	0	0	0
	2	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	0	1	1
	4	CAUSE DE REPRISE							

## Champ cause de réinitialisation

- Dans un paquet DEMANDE DE REINITIALISATION, il est à zéro.
- Dans un paquet INDICATION DE REINITIALISATION, il a la signification suivante:

	8	7	6	5	4	3	2	1
Réinitialisation par l'ETTD	0	0	0	0	0	0	0	0
Dérangement	0	0	0	0	0	0	0	1
Erreur de procédure distante	0	0	0	0	0	0	1	1
Erreur de procédure locale	0	0	0	0	0	1	0	1
Saturation du réseau	0	0	0	0	0	1	1	1

## Champ code de diagnostic

- Dans un paquet DEMANDE DE REINITIALISATION, il est à zéro
- Dans un paquet INDICATION DE REINITIALISATION, il précise une erreur de procédure locale. (Le codage sera fourni ultérieurement).

# PAQUET DE CONFIRMATION DE REINITIALISATION

		8	7	6	5	4	3	2	1
OCTET	1	0	0	0	1	NUMERO			
	2	DE VOIE LOGIQUE							
	3	0	0	0	1	1	1	1	1

## PAQUET D'INTERRUPTION

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1				
2	DE VOIE LOGIQUE							
3	0	0	1	0	0	0	1	1
4	DONNES D'INTERRUPTION DE L'UTILI- SATEUR.							

L'octet 4 est un champ de données à la disposition de l'ETTD qui l'émet. Il peut y préciser le motif de l'interruption.

## PAQUET DE CONFIRMATION D'INTERRUPTION

	8	7	6	5	4	3	2	1
OCTET 1	0	0	0	1	NUMERO			
2	DE VOIE LOGIQUE							
3	0	0	1	0	0	1	1	1



# A N N E X E    G 5

TYPE DE PAQUET	Octet 1 éléments binaires	Octet 3 éléments binaires							
	8 7 6 5	8	7	6	5	4	3	2	1
APPEL APPEL ENTRANT	0 0 0 1	0	0	0	0	1	0	1	1
COMMUNICATION ETABLIE ACCEPTÉE	0 0 0 1	0	0	0	0	1	1	1	1
DEMANDE INDICATION DE LIBERATION	0 0 0 1	0	0	0	1	0	0	1	1
CONFIRMATION DE LIBERATION	0 0 0 1	0	0	0	1	0	1	1	1
<u>DONNEES ET INTERRUPTION</u>									
DONNEES	X 0 0 1	X	X	X	X	X	X	X	0
INTERRUPTION	0 0 0 1	0	0	1	0	0	0	1	1
CONFIRMATION D'INTERRUPTION	0 0 0 1	0	0	1	0	0	1	1	1
<u>CONTROLE DE FLUX ET REINITIALISATION</u>									
RR	0 0 0 1	X	X	X	0	0	0	0	1
RNR	0 0 0 1	X	X	X	0	0	1	0	1
DEMANDE INDICATION DE REINITIALISATION	0 0 0 1	0	0	0	1	1	0	1	1
CONFIRMATION DE REINITIALISATION	0 0 0 1	0	0	0	1	1	1	1	1
<u>REPRISE</u>									
INDICATION DE REPRISE DEMANDE	0 0 0 1	1	1	1	1	1	0	1	1
CONFIRMATION DE REPRISE	0 0 0 1	1	1	1	1	1	1	1	1

X = bit à 0 ou 1

## A N N E X E    H

---

### LA PROCÉDURE X.21 ET SON ALIGNEMENT SUR X.25

=====

On trouvera dans cette annexe des renseignements concernant la procédure X.21 et son alignement sur X.25. Nous avons déjà évoqué cette question au chapitre VI paragraphe 4.

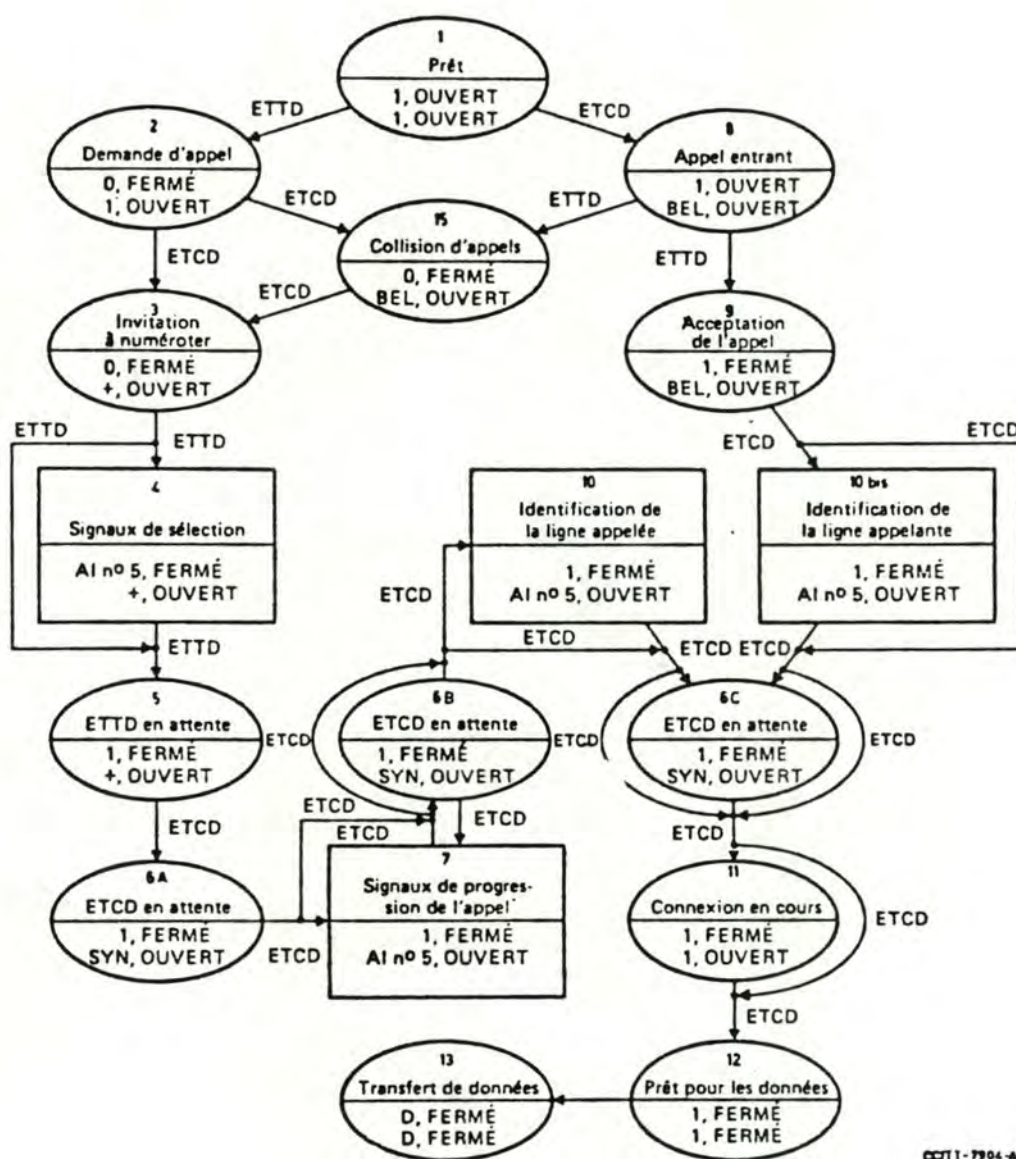
Les feuilles suivantes porteront sur:

- H.1 - les diagrammes des états de signalisation à l'interface ETTD / ETCD pour le fonctionnement synchrone. Les trois graphes évoqueront
  - la phase d'établissement de la communication
  - la phase de libération et état de repos
  - l'établissement de la phase de transfert de données.
  
- H.2 - les diagrammes de la séquence de signalisation à l'interface ETTD/ETCD pour le fonctionnement synchrone. Les deux graphes donneront l'exemple d'un:
  - appel avec aboutissement-libération
  - établissement de la communication.
  
- H.3 - les étapes de l'échange suivant X.21. Cette énumération est à mettre en rapport avec les diagrammes de cette annexe.
  
- H.4 - La correspondance entre les onze types de paquets X.25 et les phases contenues dans X.21.

Le lecteur intéressé par un développement plus important de la procédure X.21, pourra trouver des renseignements complémentaires dans l'Avis X.21 du CCITT.



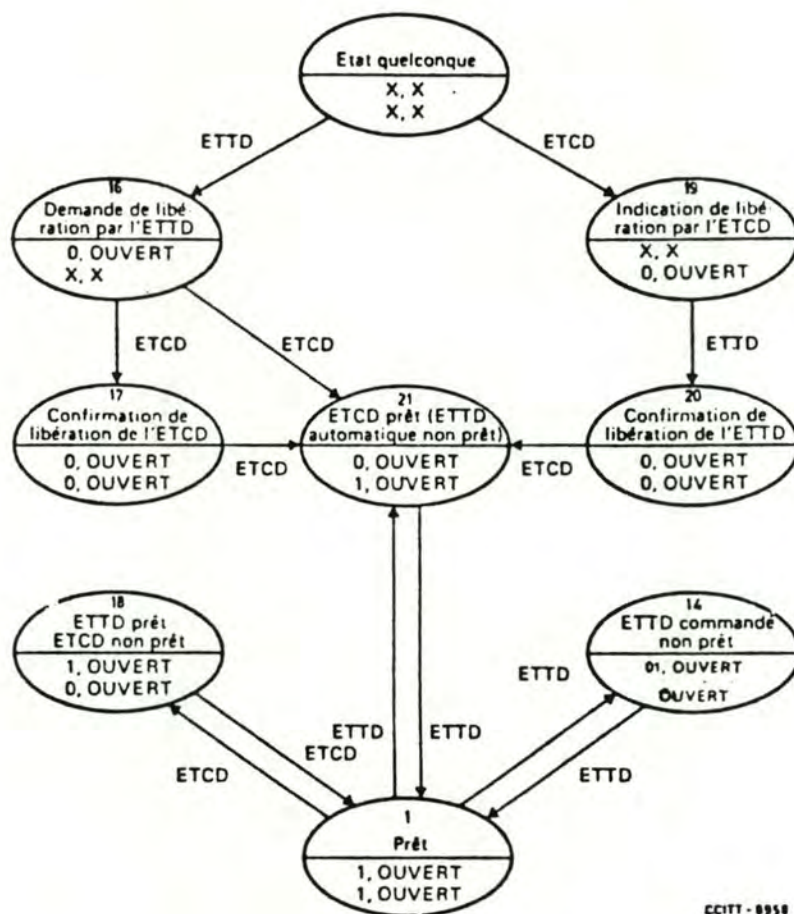
# ANNEXE H1



CCITT-7904-A

## PHASE D'ETABLISSEMENT DE LA COMMUNICATION.

Remarque : Les états 6A, 6B et 6C sont présentés dans la figure ci-dessus pour plus de commodité. Ils sont tous équivalents du point de vue fonctionnel.

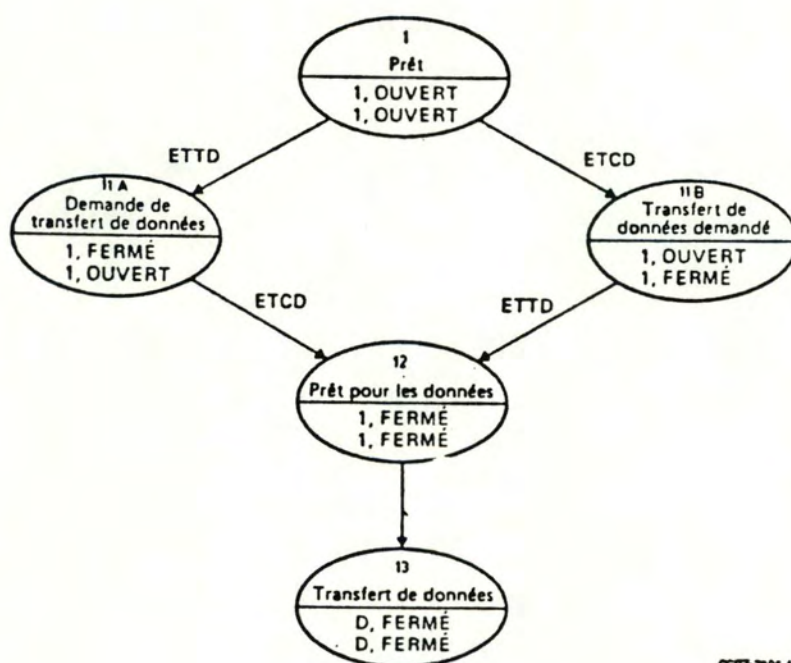


CCITT - 8958

### PHASE DE LIBERATION ET ETATS DE REPOS.

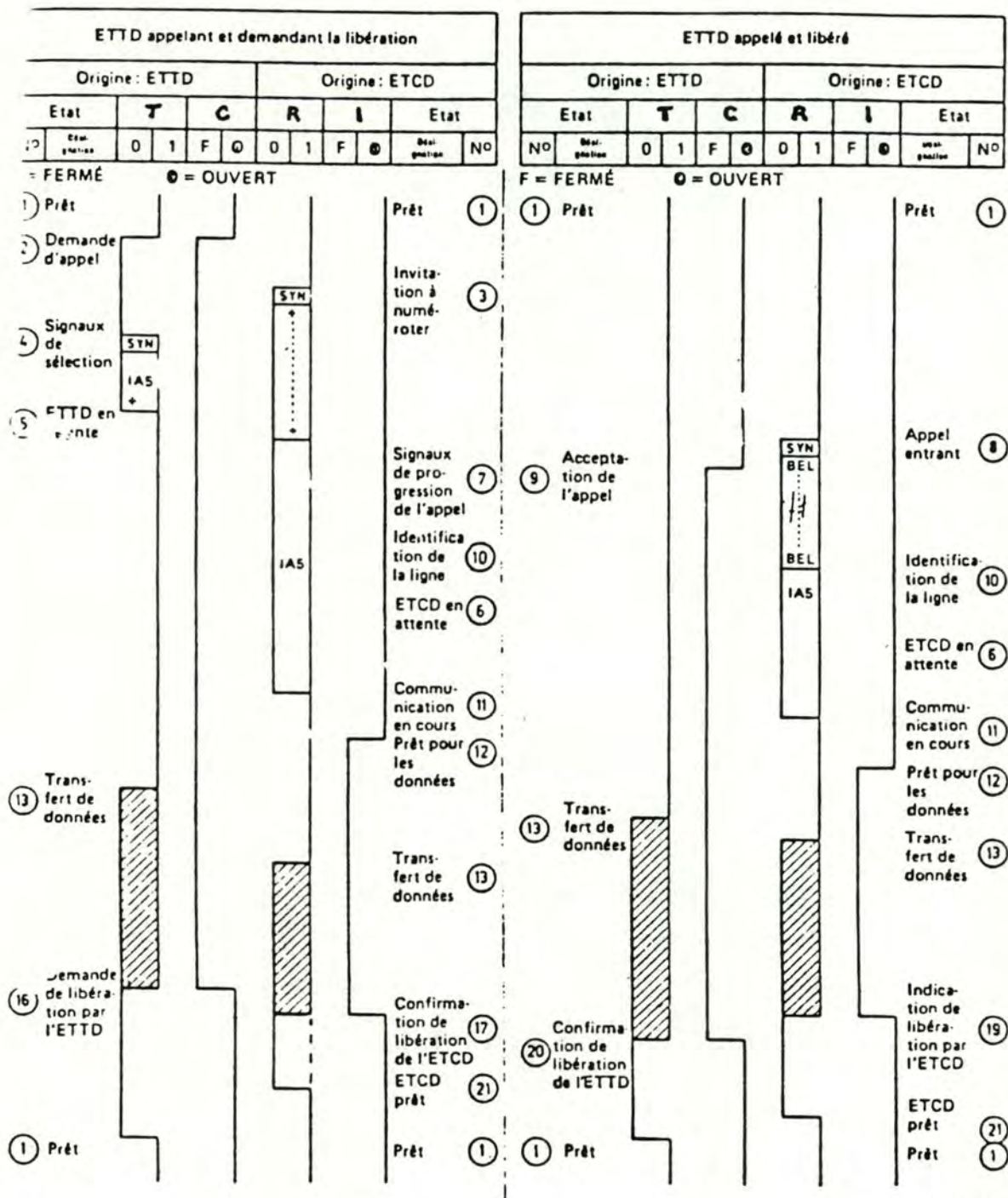
Remarque : L'ETCD peut passer à l'état 19 à partir de n'importe quel état. L'ETDD peut quant à lui passer à l'état 16 à partir de n'importe quel état.





ETABLISSEMENT DE LA PHASE DE TRANSFERT DE DONNEES.

# ANNEXE H 2



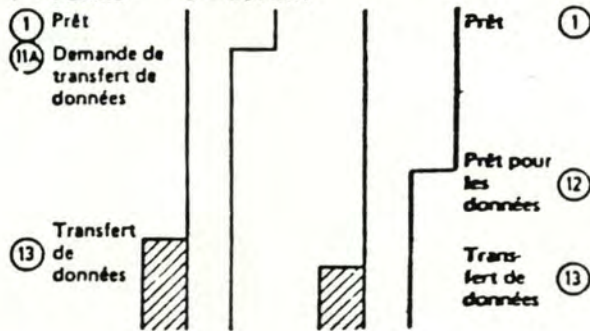
CCITT-7907-B

APPEL AVEC ABOUTISSEMENT - LIBERATION.



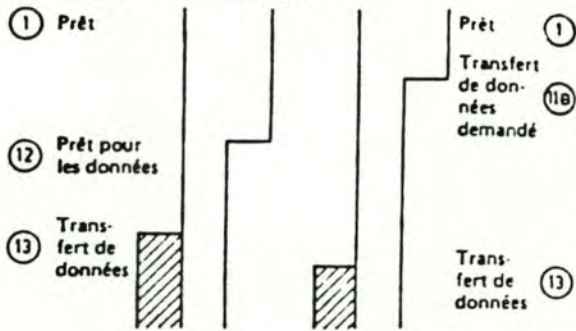
ETTD demandeur et demandant la libération											
Origine: ETTD						Origine: ETCD					
Etat		T		C		A		I		Etat	
N°	Don- nées	0	1	F	O	0	1	F	O	Don- nées	N°

F = FERMÉ      O = OUVERT



ETTD demandé et libéré											
Origine: ETTD						Origine: ETCD					
Etat		T		C		A		I		Etat	
N°	Don- nées	0	1	F	O	0	1	F	O	Don- nées	N°

F = FERMÉ      O = OUVERT



COTI-7908

# A N N E X E    H 3

X 21

	ETTD appellant	ETCD appellant	ETCD appelé	ETTD appelé
1. Prêt	X	X	X	X
2. Demande d'appel	X			
3. Invitation à numéroté		X		
4. Signaux de direction	X			
5. Attente	X			
6. Attente		X	X	
7. Signaux de progression de l'appel		X		
8. Appel entrant			X	
9. Acceptation de l'appel				X
10. Identification de la ligne		X	X	
11. Communication en cours		X	X	
12. Prêt pour les données		X	X	
13. Transfert de données	X	X	X	X
16. Demande de libération	X			
17. Confirmation de libération		X		
19. Indication de libération			X	
20. Confirmation de libération				X
21. Prêt ETCD		X	X	
11A. Transfert de données (demande)	X		X	
11B. Transfert de données demandé				



## A N N E X E    H 4

Le tableau ci-contre résume les 3 propositions décrites précédemment. Il met en regard les différents paquets de X25 face à ce que permet X21.

	<u>paquet X25</u>	<u>X21</u>
	<u>1. Etablissement</u>	
	appel	demande d'appel (2,4,5)
		CON
	appel entrant	appel entrant
MESSAGES	communication établie	identif. de la ligne (3,7,10,6)
		ACC
	communication acceptée	acceptation de l'appel + désignation de la ligne (9 et 10)
	<u>2. Libération</u>	
	demande de libération	demande de libération (16)
		DISC
SIGNALISATION	Indication de libération	(19)
	confirmation de libération	spécifiques au réseau (17)
	<u>3. Données + contrôle de flux</u>	
MESSAGES	données	SEND transfert de données (13)
	RR	non envisagé, mais on
	RNR	peut contrôler le sé- quencement au niveau des messages en introduisant S (R) et S (S).

#### 4. Réinitialisation

demande de réinitialisation

indication de réinitialisation

confirmation de réinitialisation.

spécifiques a X25, et non envisagés pour X21. On peut cependant les rapporter au niveau des segments de L4.

#### 5. Reprise

demande de reprise

indication de reprise

confirmation de reprise

spécifiques a X25 et non envisagés pour X21. On peut cependant les rapporter au niveau des segments de L4.

#### 6. Interruption

optionnel en X25 et non envisagé ici.

REMARQUE: Les chiffres entre parenthèses, indiqués pour le X21 sont ceux des diagrammes relatifs à cette procédure.



## A N N E X E    I

### ASSEMBLEUR - DÉSASSEMBLEUR DE PAQUETS: ADP

=====

On trouvera dans cette annexe des renseignements concernant l'ADP ( $\equiv$  PAD). On a évoqué la nécessité d'un ADP au chapitre 6 paragraphe 5.

Les feuilles suivantes portent sur:

- I.1 - la procédure de commande du PAD.
- I.2 - la procédure d'établissement d'un circuit virtuel commuté:
  - Etablissement à partir d'un terminal accédant par réseau téléphonique.
  - Etablissement à partir d'un terminal accédant par liaison spécialisée.
  - Etablissement par l'ETTD X.25.
- I.3 - la procédure de libération d'un circuit virtuel commuté:
  - libération par l'ETTD X.25.
  - libération par le terminal.
  - libération par le PAD ou par le réseau.
- I.4 - la procédure de transfert des données:
  - transfert de caractères
  - réinitialisation
  - interruption
  - procédure de commande d'appareil auxiliaire.

Pour obtenir de plus amples renseignements sur le PAD, il y a lieu de consulter les documents TRANSPAC.

## A N N E X E    I   1

### PROCÉDURE DE COMMANDE DU PAD

Certains terminaux ne peuvent pas s'adapter facilement à leur interface, c'est-à-dire émettre et recevoir des paquets selon le protocole X.25. C'est le cas des terminaux simples fonctionnant par caractères en mode asynchrone.

La conversion nécessaire entre les procédures propres à ces terminaux et le protocole utilisé sur les circuits virtuels est réalisée dans le commutateur d'accès du terminal, par un programme assembleur et desassembleur de paquets (PAD).

La fonction principale du PAD est d'assembler sous forme de paquets les caractères reçus du terminal et inversement, de lui délivrer sous forme de caractères les paquets qui lui sont destinés. D'autre part, il gère un dialogue avec le terminal, sous forme de message de supervision et de signalisation, permettant au terminal d'accéder aux services du circuit virtuel, établi par le PAD avec son correspondant (établissement, libération, transfert de données: ces procédures sont examinées dans les 3 annexes suivantes).

#### Commande du PAD par le terminal.

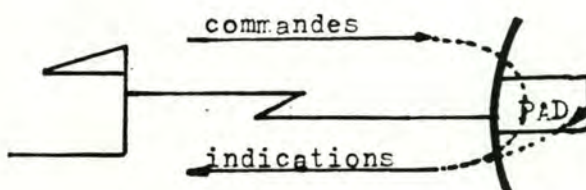
Cette procédure est utilisée par le terminal pour commander les fonctions du PAD à savoir:

- Etablissement et libération du circuit virtuel.
- Transfert de données de et vers l'ETTD X.25.
- Mise en oeuvre de services complémentaires.



Pour cela, le terminal émet des "commandes" et le PAD des "indications", pour accuser réception des commandes, y répondre et indiquer les événements entrants.

Commandes et indications sont formées d'une suite de caractères. Leur liste est donnée dans les deux tableaux suivants (fig. 1 et 2)



- Indications:

Une indication a la structure suivante:

indicateur de mise en page	/	suite mnémonique de caractères lisibles	/	indicateur de mise en page.
-------------------------------	---	---	---	-----------------------------------

L'indicateur de mise en page est: CR LF suivi de caractères NUL, en nombre variable selon la vitesse de transmission.

Pendant la transmission d'une indication, l'interface PAD terminal est en "phase d'indication". Lorsque le PAD a une indication à transmettre, il procède comme suit:

- Si l'interface est en phase de commande, il transmet d'abord l'indication associée à cette commande, puis l'autre indication.
- Sinon, le PAD transmet l'indication sans délai.

Pendant une phase d'indication, le PAD ignore tout caractère reçu du terminal.

A la fin d'une phase d'indication, si aucune communication n'est en cours, l'interface Terminal PAD passe dans l'état "PAD en attente".

- Commandes:

+ Le terminal peut émettre des commandes pendant la phase de commande. Pendant cette phase, la transmission de caractères de données au terminal est retardée.

+ Les conditions d'entrée en phase de commande sont les suivantes:

- quand le chemin d'information a été établi pour permettre au terminal d'appeler son correspondant, dans le cas où il accède par un réseau commuté téléphonique.

- sur réception par le PAD d'un caractère spécial du terminal au cours d'une phase de transfert de données, si ce dernier utilise le service complémentaire "retour en phase de commande".

+ La phase de commande est quittée sur l'une des conditions suivantes:

- Réception d'un CR

- Erreur de syntaxe

- Dépassement d'un délai de 36 secondes, après réception du premier caractère significatif de commande.

+ Dans les deux derniers cas, le PAD transmet une indication d'erreur: ERREUR.



## COMMANDES

FORMAT DE LA COMMANDE	FONCTION	INDICATION DE LA REPONSE CORRESPONDANTE
STAT	Pour demander des informations d'état relatives à tout circuit virtuel avec le terminal	LIBRE ou OCCUPE
LIB	Pour libérer un circuit virtuel	LIB CONF ou LIB ERR (dans le cas d'une erreur de procédure locale)
PAR? Liste des références des paramètres	Pour demander les valeurs courantes des paramètres cités	PAR Liste des références des paramètres et de leur valeurs courantes
SET? Liste des références des paramètres et des valeurs correspondantes	Pour demander la modification ou le positionnement des valeurs courantes des paramètres cités	PAR Liste des références des paramètres et de leurs valeurs courantes
PROF Identificateur	Pour donner aux paramètres du PAD un ensemble standard de valeurs	<p>COMMANDES N'EXIGEANT PAS DE REPONSE</p> <p>la seule confirmation de ces commandes est l'Indication d'Accusé de réception (=indicateur de mise en page CR LF )</p>
RESET	Pour réinitialiser le circuit virtuel	
INT	Pour transmettre un paquet INTERRUPTION à l'ETTD X25	
SET	Pour positionner ou modifier les valeurs des paramètres	
Annn... <sup>D</sup> xxxx..... (commande de demande d'appel nnn.... =adresse xxx.... =données d'appel)	Pour établir un circuit virtuel avec l'abonné nnnn...	

Fig. 1.



## INDICATIONS (EVENEMENTS ENTRANTS)

TEXTE DE L'INDICATION		EXPLICATION
RESET	ETTD	Indique que l'ETTD X25 a réinitialisé le circuit virtuel
	NC	Indication d'une réinitialisation d'un circuit virtuel à cause d'une saturation du réseau
COM		Indication d'appel accepté
COM	numéro de l'appelant	Indication d'appel entrant
LIB	cause de libération	Indication de libération
TRANSPAC		Indication de liaison active
ERREUR		Indication d'erreur suite à une commande que le PAD ne sait pas interpréter

### CAUSES DE LIBERATION

MNEMONIQUE	CAUSE	EXPLICATION
ETTD		L'ETTD X25 a libéré le circuit virtuel commuté
OCC	Occupation	Le numéro appelé est entièrement occupé à d'autres communications et ne peut accepter un autre appel
NC	Saturation du réseau	Les conditions de saturation du réseau empêchent momentanément l'établissement du circuit virtuel
INV	Appel non valide	Demande de service complémentaire non valable
NA	Accès interdit	Le terminal n'est pas autorisé à établir un circuit virtuel avec le numéro appelé(ex:groupe fermé)
ERR	Erreur de procédure locale	La commande reçue du terminal n'est pas correcte
RPE	Erreur de procédure à l'extrémité distante	Le circuit virtuel est libéré suite à une erreur de procédure à l'extrémité distante
NP	Numéro inconnu	
DER	Dérangement de l'appelé	1)ETTD X25 hors service 2)modem hors service 3)défaut de ligne de raccordement 4)niveau trame hors service
PAD	Libération sur invitation	Le PAD a libéré la communication sur réception d'une invitation à libérer provenant de l'ETTD X25

Fig. 2.



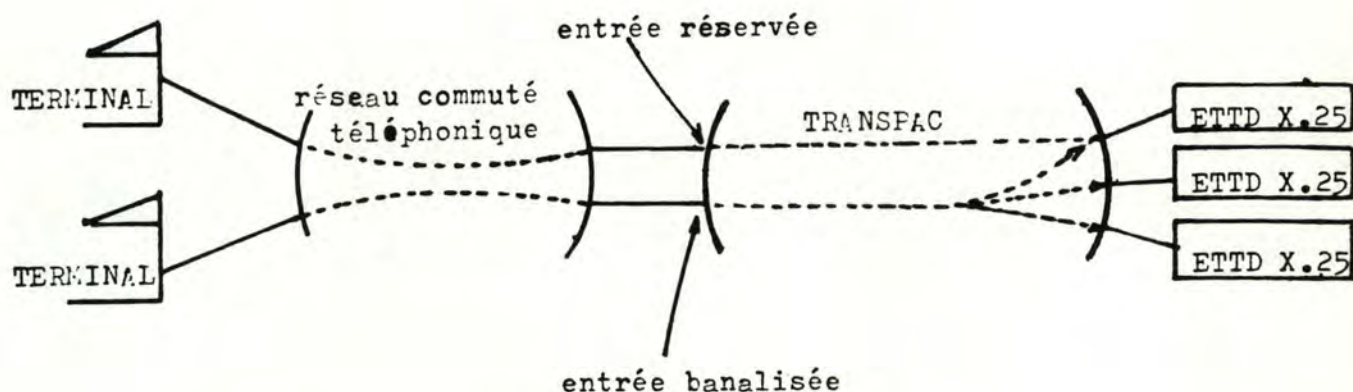
## A N N E X E    1 2

### PROCÉDURE D'ÉTABLISSEMENT D'UN CIRCUIT VIRTUEL COMMUTÉ

#### 1 - ETABLISSEMENT A PARTIR D'UN TERMINAL ACCEDANT PAR RESEAU TELEPHONIQUE

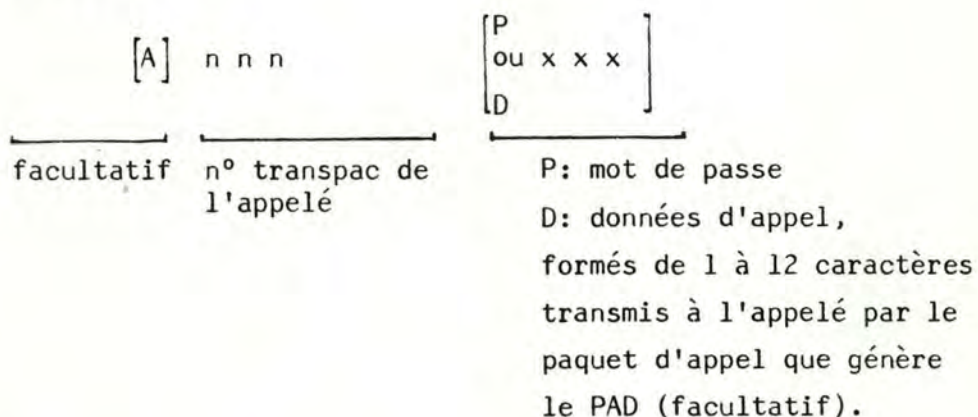
Le terminal dispose de 2 types d'entrées dans TRANSPAC:

- l'entrée réservée qui permet la mise en relation avec un abonné prédéterminé: le PAD effectue alors automatiquement l'établissement du circuit virtuel. Le terminal dispose pour y accéder, d'un numéro d'appel sur réseau commuté téléphonique propre à son correspondant.
- l'entrée banalisée qui permet la mise en relation avec un abonné quelconque. Le terminal doit fournir au PAD le numéro TRANSPAC de son correspondant. Il y a dans ce cas double numérotation.



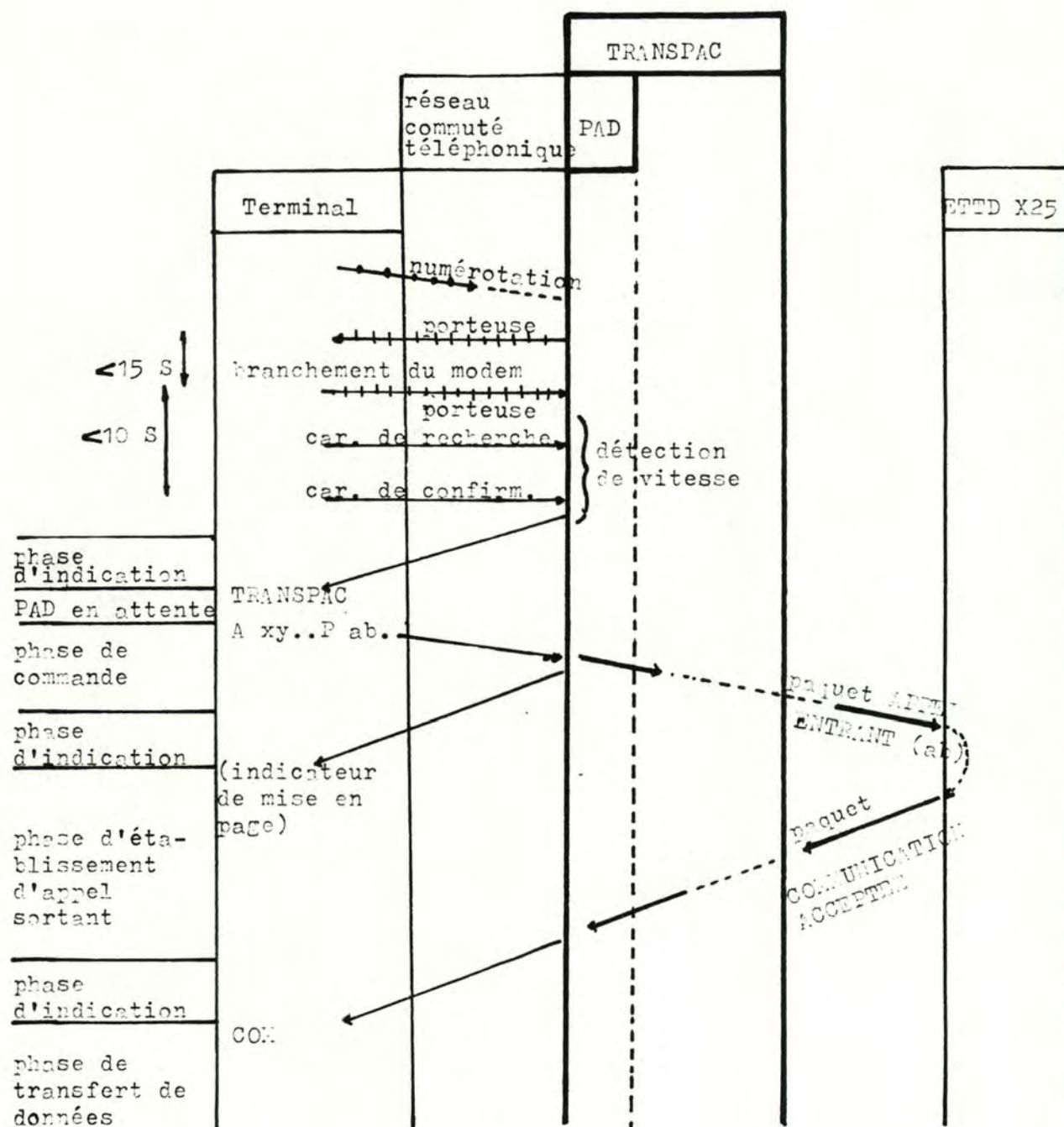
### A - Accès sur entrée banalisée (fig.3 )

- Dès que le terminal a reçu l'indication de ligne active, l'interface PAD terminal passe dans l'état "PAD en attente". Dans cet état, le PAD ignore tous les caractères compris dans les colonnes 0 et 1 du code ascii (voir annexe D ), ainsi que DEL et SP. Dès qu'il reçoit un caractère autre "significatif", il l'interprète comme un début de commande. L'interface PAD terminal passe alors en "phase de commande" .
- L'interface étant à l'état "PAD en attente", le terminal sélectionne l'ETTD appelé en transmettant une commande "demande d'appel" qui a la structure suivante:



- Le PAD confirme la réception de cette commande en transmettant l'indication d'accusé de réception, consistant en l'indicateur de mise en page (voir annexe D )
- L'interface PAD terminal passe alors en phase "d'établissement d'appel sortant". Le PAD ignore tout caractère reçu du terminal pendant cette phase.





Etablissement d'une communication par entrée banalisée

Figure 3.

- Le PAD établit le circuit virtuel avec l'abonné appelé conformément au protocole X.25.

Le code du champ de données du paquet APPEL ENTRANT que reçoit l'ETTD X.25 est le suivant:

bit	8	7	6	5	4	3	2	1
octet 1	0	0	0	0	0	0	0	1
octet 2, 3, 4	0	0	0	0	0	0	0	0
octets suivants	données d'appel du terminal							

- Le PAD prévient le terminal que la communication est établie par l'indication d'appel accepté:

COM

L'interface PAD terminal passe alors en phase de "transfert de données".

- Si la communication ne peut être acceptée, le PAD transmet l'indication de libération:

LIB + cause (voir annexe I 3)

L'interface passe alors en état PAD en attente. Si le retour à cet état a lieu plus de 5 fois sans qu'une communication n'ait été établie, le PAD libère le circuit virtuel.



B - Accès sur entrée réservée (fig. 4)

- Dès que le terminal a reçu l'indication de ligne active, le PAD entreprend l'établissement du circuit virtuel avec l'abonné appelé, dont le numéro est fixé à l'avance. L'interface PAD terminal passe alors en phase d'établissement d'appel sortant, où il ignore tout caractère reçu du terminal.

Le codage du champ de données d'appel de l'utilisateur du paquet "Appel Entrant" transmis à l'ETTD X.25 appelé est le même qu'en (a), sauf qu'il ne contient pas de données d'appel du terminal.

- Le PAD prévient le télécype que la communication est établie par l'indication d'appel accepté:

COM

L'interface PAD terminal passe alors en phase de "transfert de données".

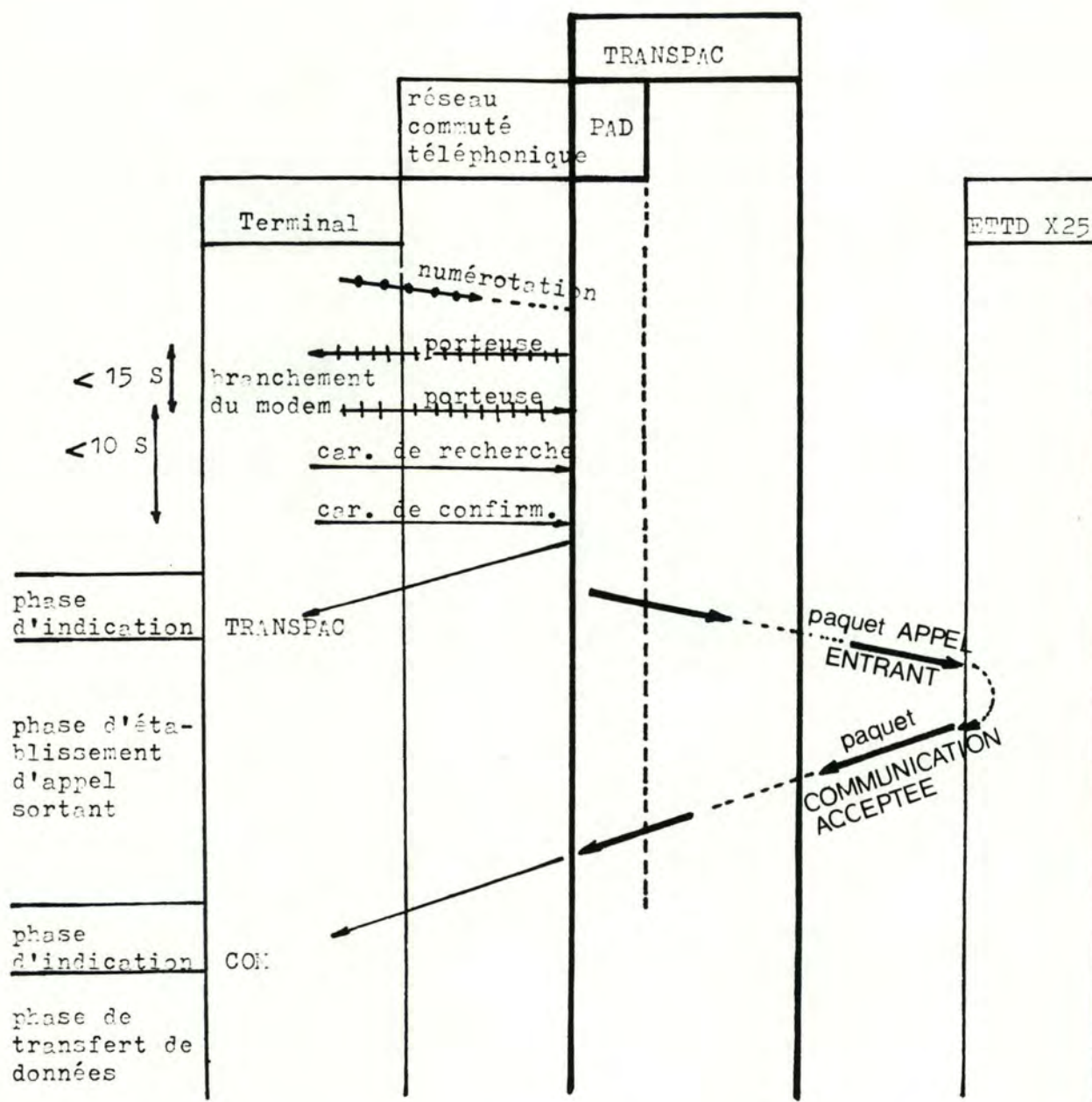
- Si la communication ne peut être établie le PAD transmet l'indication de libération:

LIB + cause

et coupe la communication téléphonique.

2 - ETABLISSEMENT A PARTIR D'UN TERMINAL ACCEDANT PAR LIAISON SPECIALISEE

Une fois que le PAD a transmis l'indication de ligne active, l'interface PAD - terminal passe dans l'état "PAD en attente". L'établissement de la communication se déroule comme pour l'accès par entrée banalisée, si ce n'est le fait que le nombre de retour en état "PAD en attente" n'est pas limité.



Etablissement d'une communication par entrée réservée

Figure 4.



### 3 - ETABLISSEMENT PAR L'ETTD X.25 (fig. 5)

L'ETTD X.25 ne peut appeler un terminal que si ce dernier est connecté par une liaison spécialisée.

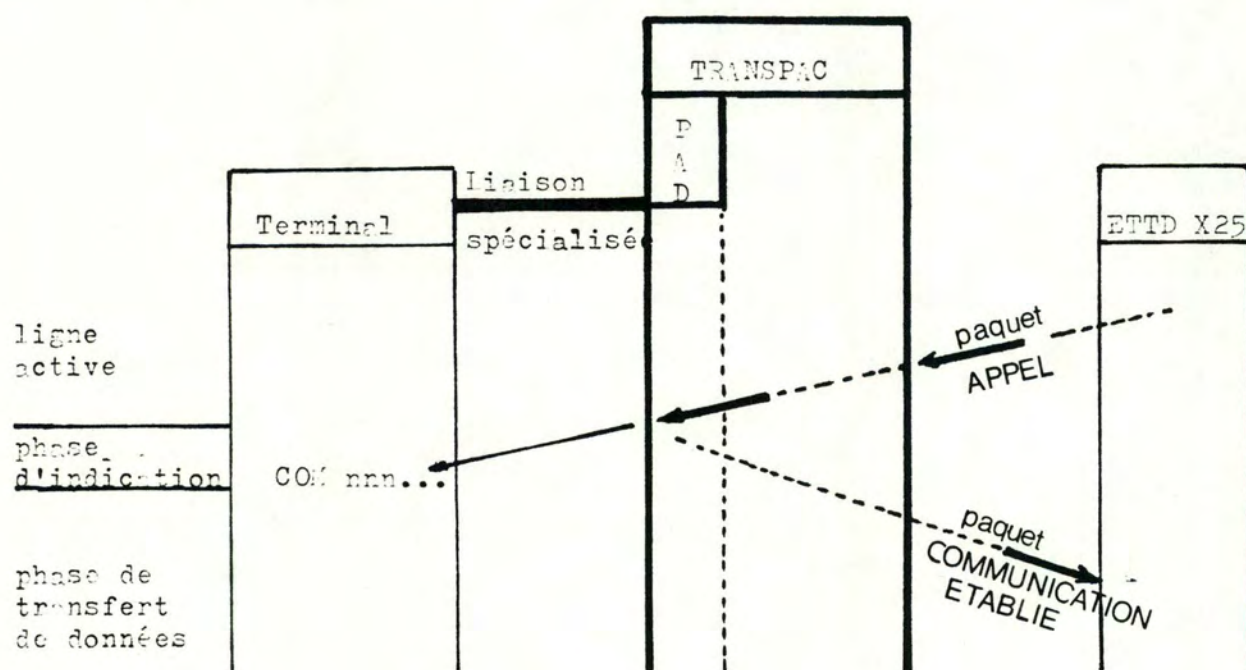
La procédure entre PAD et l'ETTD X.25 pour l'établissement du circuit virtuel commuté est conforme au protocole standard X.25.

L'interface PAD terminal doit se trouver dans l'état de ligne active. Si le PAD accepte la communication, le terminal en est averti par l'indication d'appel entrant:

COM      n  n  n  

          n° transpac de l'ETTD X.25

Le PAD n'en attend pas confirmation par le terminal.



Etablissement d'une communication par l'ETDD X.25

Figure 5.



## A N N E X E    I 3

### PROCÉDURE DE LIBÉRATION D'UN CIRCUIT VIRTUEL COMMUTÉ

#### 1 - LIBÉRATION PAR L'ETTD X.25 (fig. 6)

L'ETTD X25 peut libérer le circuit virtuel commuté, conformément au protocole standard X.25. Le PAD en avertit le terminal par l'indication de libération:

LIB ETTD

L'interface PAD terminal repasse ensuite en état "PAD en attente" si le terminal accède par entrée banalisée ou par liaison spécialisée. S'il accède par entrée réservée, le PAD libère la communication téléphonique.

L'ETTD X.25 peut demander au PAD de libérer le circuit virtuel après qu'il ait transmis au terminal toutes les données émises antérieurement par l'ETTD X.25.

Pour ce faire, l'ETTD X.25 envoie le message PAD "invitation à libération" dont l'octet type est:

bit	8	7	6	5	4	3	2	1
	0	0	0	0	0	0	0	1

Le PAD en avertit le terminal par l'indication de libération:

LIB PAD

et libère ensuite la communication téléphonique (le cas échéant). L'ETTD X.25 recevra un paquet INDICATION DE LIBÉRATION avec pour cause: "Libération par l'ETTD".

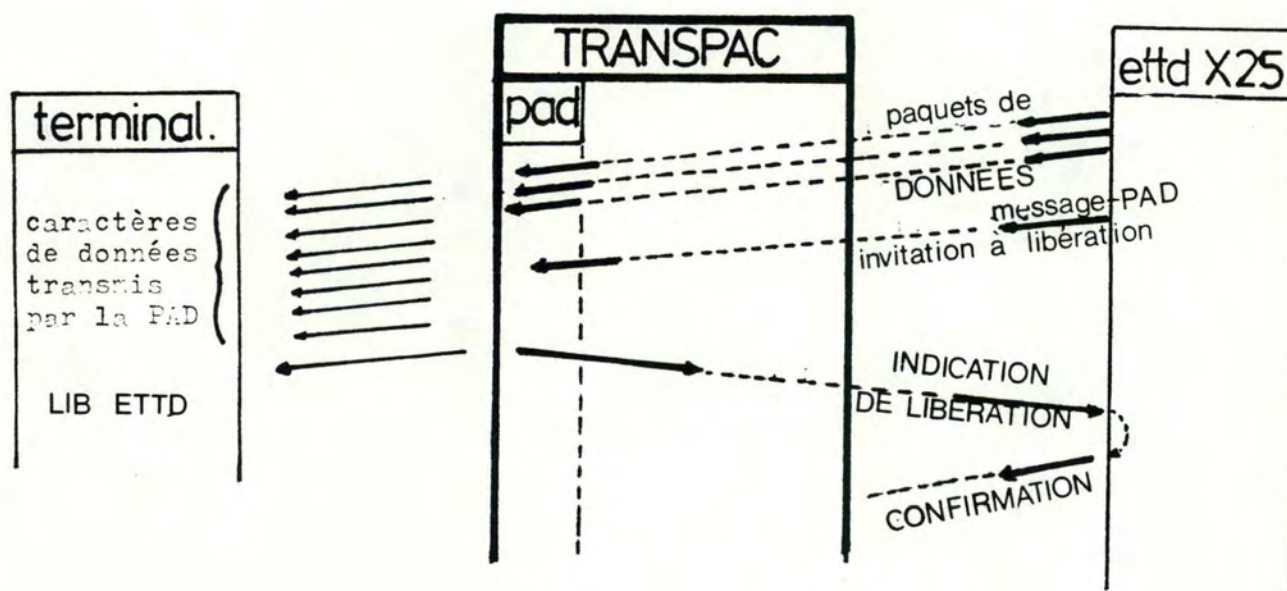


Figure 6.



## 2 - LIBERATION PAR LE TERMINAL

Lorsque le terminal se déconnecte, le PAD libère le circuit virtuel. L'ETTD X.25 reçoit alors un paquet INDICATION DE LIBERATION, avec pour cause: dérangement.

S'il a souscrit au service complémentaire de retour en phase de commande, le terminal peut demander au PAD de libérer le circuit virtuel commuté par la commande: LIB.

Le PAD la confirme par l'indication: LIB CONF.

L'ETTD X.25 recevra un paquet INDICATION DE LIBERATION avec pour cause "libération par l'ETTD".

## 3 - LIBERATION PAR LE PAD OU PAR LE RESEAU

Le terminal en est averti par une indication de libération:

LIB cause

L'ETTD X.25 reçoit une INDICATION DE LIBERATION.

## A N N E X E    I 4

### PROCÉDURE DE TRANSFERT DES DONNÉES

Ces procédures permettent la transmission de données indépendamment dans les deux sens de transmission.

Elles concernent le transfert de caractères de données, des interruptions et des réinitialisations entre le terminal et l'ETTD X.25.

#### 1 - TRANSFERT DES CARACTERES

Entre le PAD et le terminal. \_

Les caractères de données émis par le terminal sont assemblés par le PAD pour former des paquets, ayant le bit 0 qualificateur de données à 0. Le PAD émet ces paquets sous l'une des conditions suivantes:

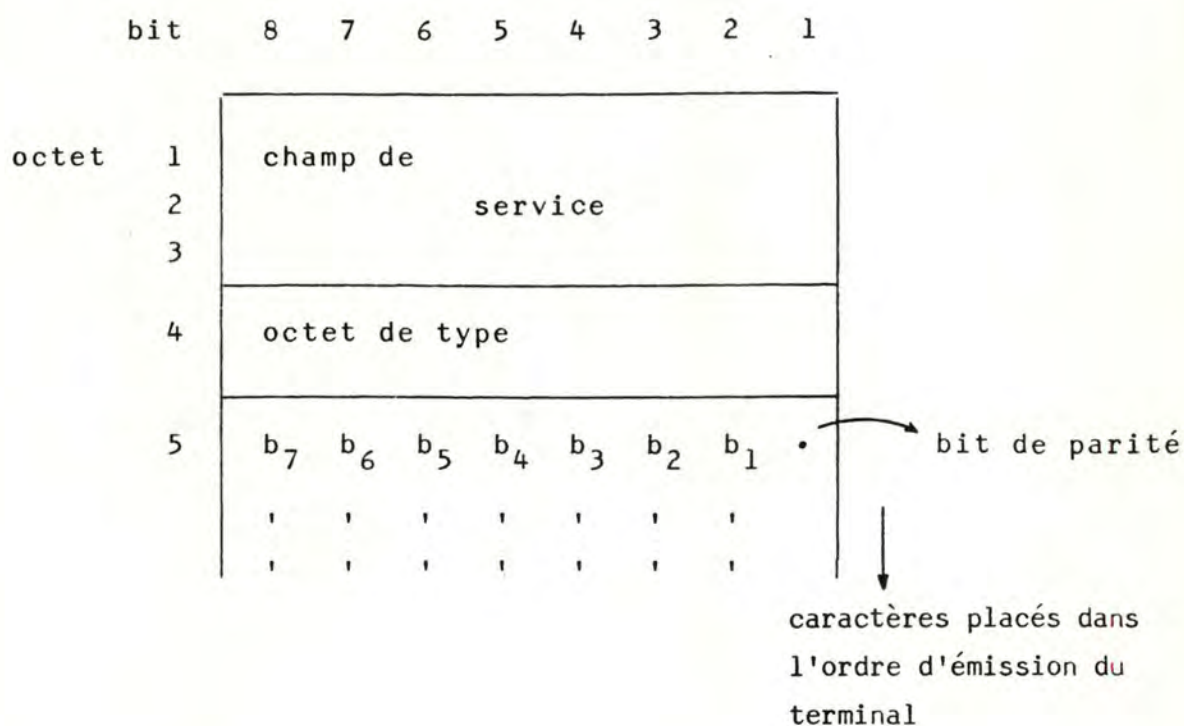
- la taille maximale du paquet est atteinte. Il est alors transmis avec le bit M (données à suivre) à 1.
- aucun caractère n'a été reçu depuis une seconde.
- réception du caractère d'envoi des données: CR.
- réception d'un break (transmission de zéro binaire pendant une durée de 50 éléments).

Toutefois, ces conditions ne pourront pas provoquer l'envoi d'un paquet vide.

Entre le PAD et l'ETTD X.25

Le codage des paquets de données transportant les caractères est le suivant:





## 2 - REINITIALISATION

Cette procédure est utilisée pour supprimer dans le réseau (et le PAD) toutes les données et interruptions relatives à la communication.

Elle se déroule entre le PAD et l'ETTD X.25 conformément au protocole standard X.25.

Le terminal peut provoquer une réinitialisation en émettant un signal "break". Il peut également le faire par la commande: RESET, s'il utilise le service complémentaire de retour en phase commande.

Le PAD lui confirme par l'indication d'accusé de réception (indicateur de mise en page CR/LF).

Le terminal sera averti d'une réinitialisation (provoquée par TRANSPAC ou l'ETTD X.25) par l'indication:

RESET mnemonique précisant la cause

L'interface PAD terminal repasse ensuite en phase de transfert de données.

### 3 - INTERRUPTION

Le PAD confirme les paquets "INTERRUPTION" transmis par l'ETTD X.25 mais n'effectue aucune autre action.

Le terminal a la possibilité de faire parvenir un paquet INTERRUPTION à l'ETTD X.25, s'il utilise le service complémentaire de retour en phase de commande. Il utilise alors la commande: INT.

Le PAD la confirme par l'indication d'accusé de réception (CR LF).

L'ETTD X.25 a la possibilité de demander au PAD de générer un signal "BREAK" destiné au terminal, par le message PAD: "Invitation à rupture". Ce message ne comporte pas de champ de paramètre.

### 4 - PROCEDURE DE COMMANDE D'APPAREIL AUXILIAIRE

Cette procédure peut être utilisée par le PAD, quand le terminal travaille avec un dispositif auxiliaire (lecteur de ruban, cassette...).

Chaque fois que le PAD est en mesure d'accepter un certain nombre de caractères du terminal, il lui transmet le caractère X-ON qui permet de mettre en marche le périphérique.

Lorsque le PAD ne désire plus en recevoir, il transmet le caractère X-OFF qui permet de l'arrêter.

Chaque fois que le PAD passe en phase de transfert de données, le PAD émet X-ON.



CCITTA N N E X E JTemporary Document No. 165  
- PART THREE -STUDY GROUP VIIOriginal : English

Geneva, 7-17 December 1982

Question(s) : 24/VII

SOURCE : Special rapporteur on Recommendations X.3, X.28 and X.29  
Mr. E. J. Blaustein - United StatesTITLE : *Assessed Recommendations X.28 and X.29*  
*(PART THREE)*

# Recommendation X.29

## PROCEDURES FOR THE EXCHANGE OF CONTROL INFORMATION AND USER DATA BETWEEN A PACKET ASSEMBLY/DISASSEMBLY FACILITY (PAD) AND A PACKET MODE DTE OR ANOTHER PAD

*(provisional, Geneva, 1977; amended, Geneva, 1980)  
(amended at Geneva, 1982)*

### Preface

The establishment in various countries of public data networks providing packet-switched data transmission services creates a need to produce standards to facilitate international interworking.

The CCITT,

*considering*

(a) that Recommendations X.1 [1] and X.2 [2] define the user classes of service and facilities in a public data network, and Recommendation X.96 [3] defines call progress signals;



(b) that Recommendation X.25 defines the interface between the DTE and the DCE for DTEs operating in the packet mode in public data networks;

(c) that Recommendation X.3 [4] defines the PAD in a public data network;

(d) that Recommendation X.28 defines the DTE/DCE interface for a start-stop mode DTE accessing the PAD in a public data network;

(e) the need to allow interworking between a packet mode DTE and a non-packet mode DTE in the packet-switched transmission service;

(f) the urgent need to allow interworking between a start-stop mode DTE in a public switched telephone network, public switched data network or a leased line and a packet mode DTE using the virtual call facility of the packet-switched transmission service;

(g) the need to allow interworking between PADs;

(h) that the packet mode DTE shall not be obliged to use the control procedures for PAD functions, but that some packet mode DTEs may wish to control specific functions of the PAD;

*unanimously declares the view that*

(1) the Recommendation X.29 procedures shall apply to the Recommendation X.25 interface between the DCE and the packet mode DTE;

(2) the Recommendation X.29 procedures may be applied for interworking between PADs;

(3) the procedures be as specified below in § 1 *Procedures for the exchange of PAD control information and user data*;

(4) the manner in which user data is transferred be as specified below in § 2 *User data transfer*;

(5) the procedures for the control of the PAD via PAD messages be as specified below in § 3 *Procedures for the use of PAD messages*;

(6) the formats of the data fields which are transferable on a virtual call be as specified below in § 4 *Formats*.

*Note 1* — For ease of understanding, this Recommendation refers to specific packet types and procedures of Recommendation X.25. When PAD to PAD interworking is considered within a national network these packet types or procedures may be represented in a different form from that used in Recommendation X.25 but will have the same operational meaning.

*Note 2* — The following items are for further study:

- the use of the permanent virtual circuit facility;
- interworking between DTEs having interfaces to different data transmission services;
- operation of non-packet mode DTEs in other than start-stop mode.

## **I Procedures for the exchange of PAD control information and user data**

**1.1** The exchange of control information and user data between a PAD and a packet mode DTE or between PADs is performed by using user data fields defined in Recommendation X.25.

**1.2** The Annex briefly describes some of the characteristics of virtual calls as defined in Recommendation X.25, as related to the PAD representation of a start-stop mode DTE to a packet mode DTE.

### **1.3 Call user data**

The call user data field of *incoming call* or *call request* packets to or from the packet mode DTE or the PAD is comprised of two fields:

- a) the protocol identifier field, and
- b) the call data field.



The protocol identifier field is used for protocol identification purposes and a call data field contains user data.

A call request packet received by the PAD, containing no call user data field, will be accepted by the PAD.

If a call data field is present, the PAD will send it, unchanged, to the start-stop mode DTE, using the call data block of the incoming call PAD service signal (see § 3.5.22, Recommendation X.28).

#### 1.4 User sequences

1.4.1 User sequences are used to exchange user data between the PAD and the packet mode DTE or a PAD.

1.4.2 User sequences are conveyed in the user data fields of complete packet sequences with  $Q = 0$ , and in both directions on a virtual call. (See Recommendation X.25.)

1.4.3 There will be only one user sequence in a complete packet sequence.

1.4.4 The operation of the PAD in respect of the transmission and reception of data packets with the D bit set to 1 is for urgent further study.

#### 1.5 PAD messages

1.5.1 PAD messages are used to exchange control information between the PAD and the packet mode DTE or PAD. A PAD message consists of a control identifier field and a message code field possibly followed by a parameter field (see § 4.4 below).

1.5.2 PAD messages are conveyed in the user data fields of complete packet sequences with  $Q = 1$  and in both directions on a virtual call. (See Recommendation X.25.)

1.5.3 There will be only one PAD message in a complete packet sequence.

1.5.4 The PAD will take into consideration a PAD message only when it has been completely received.

1.5.5 In the case where a parameter reference (see § 3 below) appears more than once in a PAD message, only the last appearance is taken into account.

1.5.6 The operation of the PAD in respect of the transmission and reception of data packets with the D bit set to 1 is for urgent further study.

### 2 User data transfer

2.1 Packets will be forwarded from the PAD when a *set*, *read*, or *set and read* PAD message is received, or under any of the other data forwarding conditions provided by the PAD (see Recommendation X.28).

2.2 The occurrence of a data forwarding condition will not cause the PAD to transmit empty data packets.

### 3 Procedures for the use of PAD messages

#### 3.1 Procedures for reading, setting, and reading and setting of PAD parameters

3.1.1 The current values of PAD parameters may be changed and read by transmitting to the PAD a *set*, *read*, or *set and read* PAD message.

3.1.2 When the PAD receives a *set*, *read* or *set and read* PAD message, any data previously received will be delivered to the start-stop mode DTE before taking action on the PAD message. The PAD will also consider the arrival of such a PAD message as a data forwarding condition.

3.1.3 The PAD will respond to a valid *read* or *set and read* PAD message by transmitting a *parameter indication* PAD message with a parameter field containing a list of the parameter references and current values, after any necessary modification, of the PAD parameters to which the PAD message received referred.

3.1.4 The PAD will not return a *parameter indication* PAD message in response to a valid *set* PAD message received.

3.1.5 Table 1/X.29 specifies the PAD's response to *set*, *set and read*, and *read* PAD messages.

3.1.6 If the character, selected by a PAD parameter value is duplicated for several different parameters the PAD will perform the function of the parameter that has the highest priority.  
(See section 3.3.2 of Recommendation X.28).



TABLE 1/X.29

PAD messages transmitted by the PAD in response to set, set and read, and read PAD messages

PAD message received by the PAD		Action upon PAD parameters	Corresponding parameter indication PAD message transmitted to the packet mode DTE (see Note 2)
Type	Parameter field		
Set	None	Reset all implemented Recommendation X.3 [4] parameters to their initial values corresponding to the initial profile (see Note 1)	None
	List of selected parameters with the desired values	Set the selected parameters to the given values: a) if no error is encountered b) if the PAD fails to modify the values of some parameters	a) None b) List of these invalid parameters with the error bit set
Set and read	None	Reset all implemented Recommendation X.3 [4] parameters to their initial values corresponding to the initial profile (see Note 1)	List all implemented Recommendation X.3 [4] parameters, and their initial values (see Note 1)
	List of selected parameters with the desired values	Set the selected parameters to the given values	List of these parameters with their new current values with the error bit set, as appropriate
Read	None	None	List all implemented Recommendation X.3 [4] parameters with their current values (see Note 1)
	List of selected parameters	None	List of these parameters with their current values

Note 1 - The procedure for setting, setting and reading, and reading parameter values for parameters not contained in Recommendation X.3 [4] is for further study.

Note 2 - The corresponding parameter indication PAD message transmitted to the packet-mode DTE when the parameter field is coded with all 0's is for further study.

### 3.2 Procedures for inviting the PAD to clear

3.2.1 The invitation to clear PAD message is used to request the PAD to clear the virtual call, after transmission to the start-stop mode DTE of all previously transmitted data.

Note - The clear indication packet, which is transmitted by the PAD after delivery of the last character to the start-stop mode DTE, will have a clearing cause field set to DTE clearing.

### 3.3 Interrupt and discard procedures

3.3.1 If parameter 7 is set to 21, the PAD will transmit an interrupt packet with all bits of the interrupt user data field set to 0 followed by an indication of break PAD message to indicate that the PAD, at the request of the start-stop mode DTE, is discarding the user sequences received. The PAD message will contain an indication in its parameter field that parameter 8 (see Recommendation X.3 [4]) has been set to 1 (discard output).



3.3.2 Before resuming data transmission to the PAD, the response to the *indication of break PAD* message shall be a *set or set and read PAD* message, indicating that parameter 3 should be set to 0 (*normal data delivery*).

Prior to sending this PAD message, any in-progress complete packet sequence being transmitted to the PAD must be terminated (with a packet that will be discarded by the PAD) in accordance with V.25 procedures.

3.3.3 If a PAD receives an *indication of break PAD* message which contains a parameter field as described in § 3.3.1 above, it will respond by transmitting a *set PAD* message as described in § 3.3.2 above and will transmit a *break* signal to the start-stop mode DTE. If a PAD receives an *indication of break PAD* message which does not contain a parameter field, it will not respond to the packet mode DTE or PAD but it will transmit a *break* signal to the start-stop mode DTE.

3.3.4 When the PAD transmits an *interrupt* packet after the receipt from the start-stop mode DTE of an *interrupt PAD command* signal or a *break* signal, when parameter 7 is set to 1, the interrupt user data field is coded in bits 8 to 1 as 00000001.

3.3.5 If the PAD receives an *interrupt* packet it will confirm it in accordance with Recommendation X.25 procedures. The PAD will not transmit the contents of the interrupt user data field to the start-stop mode DTE. The PAD will ignore the values of the interrupt user data field. It is for further study whether the coding of this field given in § 3.3.4 above causes a different response.

### 3.4 Procedure for resets

The procedures defined in Recommendation X.25 are used. The effect of the resetting procedure on the value of PAD parameter 3 is to reset its value to 0 (*normal data delivery*). The current values of all other PAD parameters are not affected.

### 3.5 Error handling procedures by the PAD

3.5.1 If the PAD receives a *set*, *read* or *set and read PAD* message containing an invalid reference to a PAD parameter, the parameter field within the *parameter indication PAD* message transmitted by the PAD will contain an indication of that fact. The remaining valid references to PAD parameters are processed by the PAD.

Possible reasons for an invalid access to a PAD parameter are:

- a) the parameter reference  
has not been implemented in the PAD
- b) the parameter value  
has not been implemented in the PAD or cannot be altered  
from the current setting
- c) the parameter is a read-only one: (*set* and *set and read PAD* messages only);
- d) the parameter follows an invalid parameter separator (see § 4.4.3.4 below).



3.5.2 The PAD will transmit an *error PAD* message containing the message code of an invalid *PAD* message received under the following conditions:

- a) if the PAD receives an unrecognizable message code;
- b) if the parameter field following a recognizable message code is incorrect or incompatible with the message code;
- c) if the parameter field following a recognizable message code has an invalid format.

3.5.3 The PAD will transmit an *error PAD* message if a *PAD* message containing less than 3 bits is received.

3.5.4 If the PAD receives an *error PAD* message it will not respond with a *PAD* message of any type. Subsequent action is for further study.

### 3.6 Procedures for inviting the PAD to reselect the called DTE

The Reselection PAD message is used by a packet-mode DTE to request the PAD to clear the virtual call, after transmission to the start-stop mode DTE of all previously transmitted data, and then to re-establish a call with another selected packet-mode DTE. The format of the Reselection PAD message is given by Figure 6/X.29.

It contains a Selection PAD command, as defined in Recommendation X.26, including the address of another DTE, possibly followed by user data or password, preceded by the character D or P.

The presence of a facility field is left for further study.

When the PAD receives the Reselection PAD message, the PAD:

- clears the virtual call established with the first packet-mode DTE;
- sends towards the start-stop mode DTE a "Redirection in progress" PAD service signal;
- enters the PAD Service Signal state.

The "Redirection in progress" PAD service signal contains a copy of the selection PAD command contained in the parameter field of the Reselection PAD message.

The user data being transmitted or not whether they are preceded by D or P.

- sends a Call Request packet destined to the packet-mode DTE the address of which is contained in the received Reselection PAD message. This Call Request packet may contain a User Data field consisting of the User Data or Password eventually contained in the received Invitation to Clear PAD message.

When the corresponding virtual call has been successfully established by the PAD the PAD indicates its establishment to the start-stop mode DTE by the mean of a Connected PAD service signal.

When the PAD has not been able to establish the corresponding virtual call, the PAD sends to the start-stop mode DTE a Clear indication PAD service signal.

## 4 Formats

### 4.1 Introduction

Bits of an octet are numbered 0 to 7 where bit 0 is the low order bit and is transmitted first. Octets of the call user data, of user sequences, of *PAD* messages and of interrupt user data are consecutively numbered starting from 1 and are transmitted in this order.

#### 4.2 Call user data format (see Figure 1/X.29)

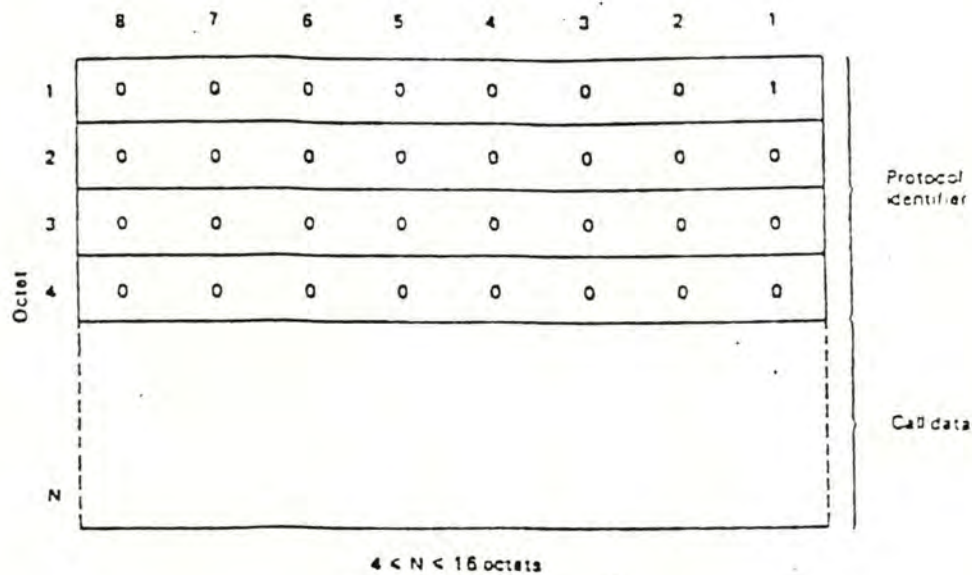


FIGURE 1/X.29  
Call user data field format

##### 4.2.1 Protocol identifier format

The protocol identifier field standardized by CCITT consists of four octets.

The first octet is coded as follows:

- bits 8 and 7 = 00 for CCITT use
  - = 01 for national use
  - = 10 reserved for international user bodies
  - = 11 for DTE-DTE use

When bits 8 and 7 are equal to 00, bits 6 to 1 are equal to 00001 for indicating PAD messages relating to the *packer assembly/disassembly* facility for the start-stop mode DTE. Other coding of bits 6 to 1 is reserved for future standardization by the CCITT. The use of octets 2, 3 and 4 is reserved and all bits are set to 0. Octets 2, 3 and 4 are reserved as a future mechanism for providing the called PAD or packet mode DTE with additional information pertinent to the calling party.

##### 4.2.2 Call data format

Octets of the call data field will contain the user characters received by the PAD from the start-stop mode DTE during the call establishment phase. The coding of these octets is similar to that of user sequences (see § 4.3 below). The call data field is limited to 12 octets (see Figure 1/X.29).

#### 4.3 User sequence format

4.3.1 The order of bit transmission from the PAD is the same as the order that bits are received from the start-stop mode DTE. The order of bit transmission to the start-stop mode DTE is the same as the order that bits are received.

4.3.2 No maximum is specified for the length of a user sequence.

#### 4.4 Control message format

4.4.1 Bits 8, 7, 6, 5 of octet 1 of a user data field of complete packet sequences with  $Q = 1$  is the *control identifier field*, used to identify the facility, such as PAD, to be controlled. The control identifier field coding for PAD messages to control a PAD for a start-stop mode DTE is 0000. Other codings of the control identifier field are reserved for future standardization.



*Note* — The possibility of extending the control identifier field is for further study.

4.4.2 When the control identifier field (see § 4.4.1 above) is set to 0000, bits 4, 3, 2, 1 of octet 1 are defined as a message code field. The *message code* field is used to identify specific types of *PAD* messages, as given in Table 2/X.29.

TABLE 2/X.29  
Type and coding of octet 1 of *PAD* messages

Type	Message code				
	Bits	4	3	2	1
Set PAD message .....	0	0	1	0	
Read PAD message .....	0	1	0	0	
Set and read PAD message .....	0	1	1	0	
Parameter indication PAD message .....	0	0	0	0	
Invitation to clear PAD message .....	0	0	0	1	
Indication of break PAD message .....	0	0	1	1	
Error PAD message .....	0	1	0	1	

*Note* — The possibility of extending the message code field is for further study.

4.4.3 All *PAD* messages consist of a control identifier field (bits 8, 7, 6, 5 of octet 1 equal to 0000) and a message code (bits 4, 3, 2, 1 of octet 1).

*Set*, *read*, *set and read* and *parameter indication PAD* messages consist of octet 1 which may be followed by one or more parameter fields. Each parameter field consists of a parameter reference octet and a parameter value octet.

The parameter value octets of the *read PAD* message contain the value 0.

The *error PAD* message consists of octet 1 and one or two octets giving the reason for the error.

The *indication of break PAD* message consists of octet 1 which may be followed by a parameter field. The parameter field, if present, consists of a parameter reference octet followed by a parameter value octet.

The *invitation to clear PAD* message consists of octet 1 only.

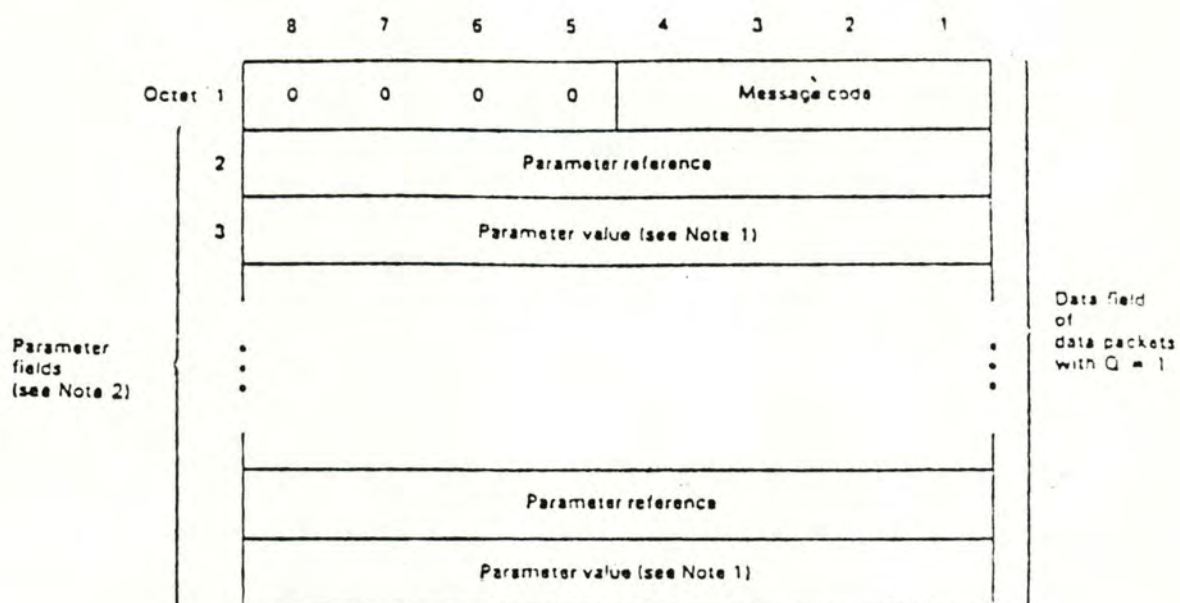
4.4.4 The maximum length of *PAD* message is for further study.

4.4.5 *Parameter field for set, read, set and read, and parameter indication PAD* messages (see Figure 2/X.29)

The parameter field of these *PAD* messages, when present, will consist of successive parts of reference fields and value fields. Each one of these fields will be one octet long.

4.4.5.1 A reference field consists of a parameter reference, identified as a decimal number in Recommendation X.3 [4], and is binary coded in bits 7 to 1, where bit 1 is the low order bit. Reference fields need not be ordered by increasing parameter reference numbers.

4.4.5.2 In *PAD* messages received by the PAD, bit 8 of each reference field will be ignored. In *parameter indication PAD* messages, bit 8 of each reference field set to 1 will indicate an invalid access to the referred parameter as described in § 3.5 above.



Message code: 0010 - Set  
 0100 - Read  
 0110 - Set and read  
 0000 - Parameter indication

Note 1 - These octets contain all 0s in read PAD messages.

Note 2 - Parameter field need not be present (see Table 1/X.29).

FIGURE 1/X.29

Set, read, set and read, and parameter indication PAD message format



4.4.5.3 A parameter value field consists of a value of the parameter reference, identified as a decimal number in Recommendation X.3 [4], and is binary coded in bits 8 to 1, where bit 1 is the low order bit. Value fields in *read PAD* messages are coded as all binary 0s. In *set* and *set and read PAD* messages, they will indicate the requested values of parameters. In *parameter indication PAD* messages, they will indicate the current values of PAD parameters, after modification if any. If bit 8 (error bit) is set to 1 in the preceding octet (i.e., the parameter reference field), the parameter value field will indicate the reason for the error, as given in Table 3/X.29.

The code 111111 (decimal 127) in bits 7 to 1 of the reference field will be used for the extension of this field. Such coding will indicate that there is another octet following. The following octet is coded with the parameter reference of Recommendation X.3 [4] minus 127.

Table 3/X.29

### Coding of Parameter Value Field in case of Error

Error Type	Parameter Value Field Code							
	Bits							
	8	7	6	5	4	3	2	1
1. No additional information	0	0	0	0	0	0	0	0
2. The parameter reference does not exist or has not been implemented in the PAD	0	0	0	0	0	0	0	1
3. The parameter value is invalid or has not been implemented in the PAD	0	0	0	0	0	0	1	0
4. The parameter value cannot be altered from the current setting	0	0	0	0	0	0	1	1
5. The parameter is read-only	0	0	0	0	0	1	0	0
6. The parameter follows an invalid parameter separator.	0	0	0	0	0	1	0	1

Note: The value 0 is mandatory, other values are optional.

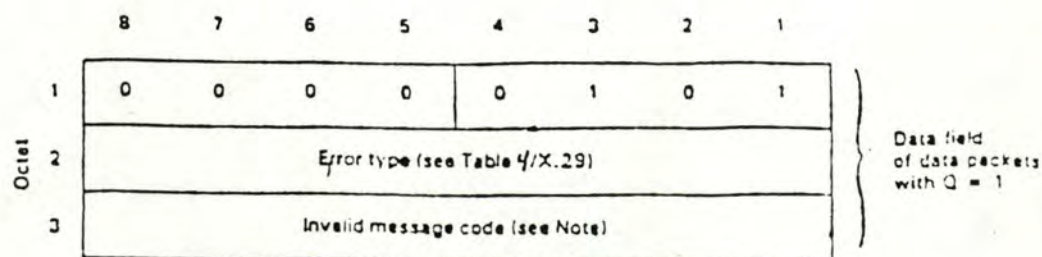
4.4.5.4 To provide for the existence of parameters not standardized by CCITT, provision is made to indicate that these parameters are to be *set*, *set and read* or *read* in appropriate *PAD* messages.

To indicate the separation between parameters listed in Recommendation X.3 [4] and any others implemented nationally or only locally, the parameter reference field in the appropriate *PAD* messages is set to 00000000 and the parameter value field is also set to 00000000.

It is for further study whether or not this mechanism is also used in *parameter indication PAD* messages to distinguish national or local parameters and their values.

Note — It is recommended that packet mode DTEs use only the parameters defined in Recommendation X.3 [4] when communicating with a PAD in a different country or network.

#### 4.4.6 Parameter field for error PAD messages (see Figure 3/X.29)



Note — Does not occur for error type 00000000.

FIGURE 3/X.29  
Error PAD message format

4.4.6.1 Octet 2 of the error PAD message will be coded as shown in Table 4/X.29.

TABLE 4/X.29  
Coding and meaning of octet 2 of error PAD messages

Case	Meaning	Coding								
		Bits	8	7	6	5	4	3	2	1
a	Received <i>PAD</i> message contained less than eight bits		0	0	0	0	0	0	0	0
b	Unrecognized message code in received <i>PAD</i> message		0	0	0	0	0	0	0	1
c	Parameter field format of received <i>PAD</i> message was incorrect or incompatible with message code		0	0	0	0	0	0	1	0
d	Received <i>PAD</i> message did not contain an integral number of octets		0	0	0	0	0	0	1	1

4.4.6.2 In cases b, c and d in Table 4/X.29, octet 3 of an error PAD message will contain the message code of the received PAD message.

#### 4.4.7 Parameter field for indication of break PAD messages (see Figure 4/X.29)

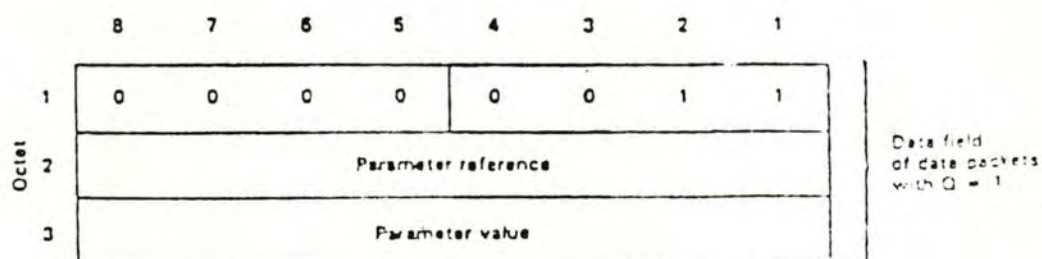


FIGURE 4/X.29  
Indication for break PAD message format



4.4.7.1 When transmitted by the packet mode DTE, this PAD message may either contain no parameter field, or contain a parameter field as described in § 4.4.7.2 below.

4.4.7.2 When transmitted by the PAD, the parameter field will contain two octets (i.e. one reference field and one value field) and will be coded as follows: the reference field will be coded 00001000 (indicating parameter 8) and the value field will be coded 00000001 (indicating decimal 1).

4.4.8 *Parameter field for invitation to clear PAD message* (see Figure 5/X.29)

	8	7	6	5	4	3	2	1	
Octet 1	0	0	0	0	0	0	0	1	Data field of data packets with Q = 1

FIGURE 5/X.29

Invitation to clear PAD message format

This PAD message will contain no parameter field.

4.4.9 Reselection PAD message format

*The format of this message is for further study.*

#### ANNEX A

(to Recommendation X.29)

#### Characteristics of virtual calls and Recommendation X.25 as related to the PAD representation of a start-stop mode DTE to a packet mode DTE

##### A.1 *General interface characteristics*

A.1.1 The mechanical, electrical, functional and procedural characteristics to activate, maintain and deactivate the physical link between the DTE and the DCE will be in accordance with the physical level procedures of Recommendation X.25.

A.1.2 The link access procedure for data interchange across the link between the DTE and DCE will be in accordance with the link level procedures of Recommendation X.25.

A.1.3 The packet format and control procedures for the exchange of packets containing control information and user data between the DTE and the DCE will be in accordance with the packet level procedures of Recommendation X.25.

##### A.2 *Interface procedures for virtual call control*

A.2.1 Incoming calls are indicated to the packet mode DTE as specified in Recommendation X.25. Call requests are indicated by the packet mode DTE as specified in Recommendation X.25. Any use of optional user facilities are indicated in accordance with § 7 of Recommendation X.25.

A.2.2 The default throughput classes used by the PAD are determined by the data rates of the start-stop mode DTE (where exact correspondence is not obtained, the next higher throughput class is used).

A.2.3 The PAD and the packet mode DTE will use the clearing procedures specified in §§ 4.1.7, 4.1.8 and 4.1.9 of Recommendation X.25.

##### A.3 *Interface procedures for data transfer*

A.3.1 Data transfer on a virtual call can only take place in the data transfer state and when flow control permits (see § 4.4 of Recommendation X.25). The same is true for the transfer of interrupt packets (see § 4.5 of Recommendation X.25).



A.3.2 *Interrupt* packets transmitted by the packet mode DTE will be confirmed by the PAD following the procedures in Recommendation X.25.

A.3.3 The reset procedure may be used by the packet mode DTE or the PAD, to re-initialize the virtual call and will conform to the procedures described in § 4.4.3 of Recommendation X.25.

A.3.4 A reset of the virtual call originated by the packet mode DTE or due to network congestion may be indicated by the PAD to the start-stop mode DTE.

A.3.5 A reset procedure initiated by the PAD may be due either to:

- a) the receipt at the PAD of a request to reset from the non-packet mode DTE. The resetting cause contained in the *reset indication* packet will be *DTE reset*; or
- b) a PAD or network failure.

A.3.6 If bit 7 of octet 1 in the *call connected* packet received by the PAD is 0, the PAD will set the D bit to 0 in all transmitted *data* packets. If bit 7 of octet 1 in the *call connected* packet received by the PAD is 1, the PAD is permitted to use the D bit procedure in *data* packets.

For calls received by the PAD with bit 7 of octet 1 in the *incoming call* packet set to 0, the PAD will set bit 7 of octet 1 in the *call accepted* packet to 0 and will set the D bit in transmitted *data* packets to 0.

Pending further study, and in the absence of bilateral agreement between Administrations (used in conjunction with the D bit modification facility), the following applies:

If the *incoming call* packet received by the PAD has bit 7 of octet 1 set to 1, the PAD may set bit 7 of octet 1 of the *call accepted* packet to 1 and, if so, may use the D bit procedure in *data* packets.

Calls originated by the PAD will set bit 7 of octet 1 in *call request* packets to 0. The called DTE can indicate if it requires the support of the D bit procedure by setting bit 7 of octet 1 of *call accepted* packets to 1.

PAD procedures associated with the Delivery Confirmation (D) bit (see § 4.3.3 of Recommendation X.25) are for further study in particular in conjunction with the further study on non-packet mode DTE other than start-stop.

## A.4 *Virtual call characteristics*

### A.4.1 *Resetting*

A.4.1.1 There may be a loss of data characters in any case of reset, as stated in Recommendation X.25. Characters generated by either of the DTEs prior to the *reset* indication or confirmation will not be delivered to the other DTE after the *reset* indication or confirmation.

### A.4.2 *Interrupt transfer*

A.4.2.1 An *interrupt* packet is always delivered at or before the point in the data packet stream at which it was generated.

### A.4.3 *Call clearing*

Data transmitted immediately before a *clear request* packet is sent may be overtaken within the network by the *clear request* packet and subsequently be destroyed.



## References

- [1] CCITT Recommendation *International user classes of service in public data networks*, Vol. VIII, Fascicle VIII.2, Rec. X.1.
- [2] CCITT Recommendation *International user services and facilities in public data networks*, Vol. VIII, Fascicle VIII.2, Rec. X.2.
- [3] CCITT Recommendation *Call progress signals in public data networks*, Vol. VIII, Fascicle VIII.3, Rec. X.96.
- [4] CCITT Recommendation *Packet assembly/disassembly facility (PAD) in a public data network*, Vol. VIII, Fascicle VIII.2, Rec. X.3.